

LAMPIRAN

Source Code Aplikasi web

1. Source Code Account

```
namespace SIBMT_INSANMADANI.Controllers
{
    [Authorize]
    public class AccountController : Controller
    {
        private ApplicationSignInManager
        _signInManager;
        private ApplicationUserManager _userManager;

        public AccountController()
        {
        }

        public AccountController(ApplicationUserManager
        userManager, ApplicationSignInManager signInManager )
        {
            UserManager = userManager;
            SignInManager = signInManager;
        }

        public ApplicationSignInManager SignInManager
        {
            get
            {
                return _signInManager ??
                HttpContext.GetOwinContext().Get<ApplicationSignInManag
                er>();
            }
            private set
            {
                _signInManager = value;
            }
        }

        public ApplicationUserManager UserManager
        {
            get
            {
                return _userManager ??
                HttpContext.GetOwinContext().Get<Application
                UserManager>();
            }
            private set
            {
                _userManager = value;
            }
        }

        // GET: /Account/Login
        [AllowAnonymous]
        public ActionResult Login(string returnUrl)
        {
            ViewBag.ReturnUrl = returnUrl;
            return View();
        }

        // POST: /Account/Login
        [HttpPost]
        [AllowAnonymous]
        [ValidateAntiForgeryToken]
        public async
        Task<ActionResult> Login(LoginViewModel model, string
        returnUrl)
        {
            if (!ModelState.IsValid)
            {
                TempData["Success"] = "Yes";
                return View(model);
            }

            // This doesn't count login failures
            towards account lockout
            // To enable password failures to trigger
            account lockout, change to shouldLockout: true
            var result = await
            SignInManager.PasswordSignInAsync(model.UserName,
            model.Password, model.RememberMe, shouldLockout:
            false);

            switch (result)
            {
                case SignInStatus.Success:
                    return RedirectToLocal(returnUrl);
                case SignInStatus.LockedOut:
                    return View("Lockout");
                case SignInStatus.RequiresVerification:
                    return RedirectToAction("SendCode",
                    new { ReturnUrl = returnUrl, RememberMe =
                    model.RememberMe });
                case SignInStatus.Failure:
                default:
                    TempData["Success"] = "";
                    //ModelState.AddModelError("",
                    "Invalid login attempt.");
                    return View(model);
            }

            // GET: /Account/VerifyCode
            [AllowAnonymous]
            public async Task<ActionResult>
            VerifyCode(string provider, string returnUrl, bool
            rememberMe)
            {
                // Require that the user has already logged
                in via username/password or external login
                if (!await
                SignInManager.HasBeenVerifiedAsync())
                {
                    return View("Error");
                }
                return View(new VerifyCodeViewModel {
                Provider = provider, ReturnUrl = returnUrl, RememberMe
                = rememberMe });
            }

            // POST: /Account/VerifyCode
            [HttpPost]
            [AllowAnonymous]
            [ValidateAntiForgeryToken]
            public async Task<ActionResult>
            VerifyCode(VerifyCodeViewModel model)
            {
                if (!ModelState.IsValid)
                {
                    return View(model);
                }

                // The following code protects for brute
                force attacks against the two factor codes.
                // If a user enters incorrect codes for a
                specified amount of time then the user account
                // will be locked out for a specified
                amount of time.
                // You can configure the account lockout
                settings in IdentityConfig
                var result = await
                SignInManager.TwoFactorSignInAsync(model.Provider,
                model.Code, isPersistent: model.RememberMe,
                rememberBrowser: model.RememberBrowser);
                switch (result)
                {
                    case SignInStatus.Success:
                        return
                        RedirectToLocal(model.ReturnUrl);
                    case SignInStatus.LockedOut:
                        return View("Lockout");
                    case SignInStatus.Failure:
                    default:
                        ModelState.AddModelError("",
                        "Invalid code.");
                        return View(model);
                }
            }
        }
    }
}
```

```

// GET: /Account/Register
[Authorize(Roles = "Admin",Users = "SuperAdmin")]
public ActionResult Register()
{
    return View();
}

//
// POST: /Account/Register
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Register(RegisterViewModel model)
{
    ApplicationDbContext context = new
    ApplicationDbContext();

    var roleManager = new
    RoleManager<IdentityRole>(new
    RoleStore<IdentityRole>(context));
    var UserManager = new
    UserManager<ApplicationUser>(new
    UserStore<ApplicationUser>(context));

    if (ModelState.IsValid)
    {
        //var user = new ApplicationUser {
        UserName = model.Email, Email = model.Email };
        var user = new ApplicationUser
        {
            UserName = model.UserName,
            Email = model.Email
        };

        var result = await
        UserManager.CreateAsync(user, model.Password);
        user =
        UserManager.FindByEmail(model.Email);
        //result =
        UserManager.SetLockoutEnabled(user.Id, false);

        //Create Role Pengelola if it does not
        exist
        var role =
        roleManager.FindByName("Pengelola");
        if (role == null)
        {
            role = new IdentityRole("Pengelola");
            var roleresult =
            roleManager.Create(role);
        }

        // Add user Pengelola to Role Pengelola
        if not already added
        var rolesForUser =
        UserManager.GetRoles(user.Id);
        if (!rolesForUser.Contains(role.Name))
        {
            result =
            UserManager.AddToRole(user.Id, role.Name);
        }

        if (result.Succeeded)
        {
            //await
            SignInManager.SignInAsync(user, isPersistent:false,
            rememberBrowser:false);

            // For more information on how to
            enable account confirmation and password reset please
            visit http://go.microsoft.com/fwlink/?LinkID=320771
            // Send an email with this link
            // string code = await
            UserManager.GenerateEmailConfirmationTokenAsync(user.Id);
            // var callbackUrl =
            Url.Action("ConfirmEmail", "Account", new { userId =
            user.Id, code = code }, protocol: Request.Url.Scheme);
            // await
            UserManager.SendEmailAsync(user.Id, "Confirm your
            account", "Please confirm your account by clicking a
            href=\"\" + callbackUrl + \"\">here</a>");

            return RedirectToAction("Create",
            "Pengelolas", new { id = user.Id });
        }
        AddErrors(result);
    }
    // If we got this far, something failed,
    redisplay form
}

```

```

TempData["Success"] = "";
return View(model);
}

[Authorize(Roles = "Pengelola")]
public ActionResult RegisterNasabah()
{
    return View();
}
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
RegisterNasabah(RegisterViewModel model)
{
    ApplicationDbContext context = new
    ApplicationDbContext();

    var roleManager = new
    RoleManager<IdentityRole>(new
    RoleStore<IdentityRole>(context));
    var UserManager = new
    UserManager<ApplicationUser>(new
    UserStore<ApplicationUser>(context));

    if (ModelState.IsValid)
    {
        var user = new ApplicationUser
        {
            UserName = model.UserName,
            Email = model.Email
        };

        var result = await
        UserManager.CreateAsync(user, model.Password);
        user =
        UserManager.FindByEmail(model.Email);

        //Create Role Nasabah if it does not
        exist
        var role =
        roleManager.FindByName("Nasabah");
        if (role == null)
        {
            role = new IdentityRole("Nasabah");
            var roleresult =
            roleManager.Create(role);
        }

        // Add user Nasabah to Role Nasabah if
        not already added
        var rolesForUser =
        UserManager.GetRoles(user.Id);
        if (!rolesForUser.Contains(role.Name))
        {
            result =
            UserManager.AddToRole(user.Id, role.Name);
        }

        if (result.Succeeded)
        {
            return RedirectToAction("Create",
            "Anggotas", new { id = user.Id });
            //return RedirectToAction("Index",
            "Home");
        }
        AddErrors(result);
    }
    // If we got this far, something failed,
    redisplay form
    TempData["Success"] = "";
    return View(model);
}

//
// GET: /Account/ConfirmEmail
[AllowAnonymous]
public async Task<ActionResult>
ConfirmEmail(string userId, string code)
{
    if (userId == null || code == null)
    {
        return View("Error");
    }
    var result = await
    UserManager.ConfirmEmailAsync(userId, code);
    return View(result.Succeeded ? "ConfirmEmail"
    : "Error");
}

```

```

//
// GET: /Account/ForgotPassword
[AllowAnonymous]
public ActionResult ForgotPassword()
{
    return View();
}

//
// POST: /Account/ForgotPassword
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
ForgotPassword(ForgotPasswordViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = await
UserManager.FindByEmailAsync(model.Email);
        if (user == null || !(await
UserManager.IsEmailConfirmedAsync(user.Id)))
        {
            // Don't reveal that the user does
not exist or is not confirmed
            return
View("ForgotPasswordConfirmation");
        }

        // For more information on how to enable
account confirmation and password reset please visit
http://go.microsoft.com/fwlink/?LinkID=320771
        // Send an email with this link
        // string code = await
UserManager.GeneratePasswordResetTokenAsync(user.Id);
        // var callbackUrl =
Url.Action("ResetPassword", "Account", new { userId =
user.Id, code = code }, protocol: Request.Url.Scheme);

        // await
UserManager.SendEmailAsync(user.Id, "Reset Password",
"Please reset your password by clicking <a href=\"\" +
callbackUrl + \">here</a>");
        // return
RedirectToAction("ForgotPasswordConfirmation",
"Account");
    }

    // If we got this far, something failed,
redisplay form
    return View(model);
}

//
// GET: /Account/ForgotPasswordConfirmation
[AllowAnonymous]
public ActionResult ForgotPasswordConfirmation()
{
    return View();
}

//
// GET: /Account/ResetPassword
[AllowAnonymous]
public ActionResult ResetPassword(string code)
{
    return code == null ? View("Error") : View();
}

//
// POST: /Account/ResetPassword
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
ResetPassword(ResetPasswordViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
    var user = await
UserManager.FindByEmailAsync(model.Email);
    if (user == null)
    {
        // Don't reveal that the user does not
exist
        return
RedirectToAction("ResetPasswordConfirmation", "Account");
    }
}

```

```

var result = await
UserManager.ResetPasswordAsync(user.Id, model.Code,
model.Password);
if (result.Succeeded)
{
    return
RedirectToAction("ResetPasswordConfirmation", "Account");
}
AddErrors(result);
return View();
}

//
// GET: /Account/ResetPasswordConfirmation
[AllowAnonymous]
public ActionResult ResetPasswordConfirmation()
{
    return View();
}

//
// POST: /Account/ExternalLogin
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public ActionResult ExternalLogin(string
provider, string returnUrl)
{
    // Request a redirect to the external login
provider
    return new ChallengeResult(provider,
Url.Action("ExternalLoginCallback", "Account", new {
ReturnUrl = returnUrl }));
}

//
// GET: /Account/SendCode
[AllowAnonymous]
public async Task<ActionResult> SendCode(string
returnUrl, bool rememberMe)
{
    var userId = await
SignInManager.GetVerifiedUserIdAsync();
    if (userId == null)
    {
        return View("Error");
    }
    var userFactors = await
UserManager.GetValidTwoFactorProvidersAsync(userId);
    var factorOptions =
userFactors.Select(purpose => new SelectListItem { Text =
purpose, Value = purpose }).ToList();
    return View(new SendCodeViewModel { Providers
= factorOptions, ReturnUrl = returnUrl, RememberMe =
rememberMe });
}

// POST: /Account/SendCode
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
SendCode(SendCodeViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View();
    }

    // Generate the token and send it
    if (!await
SignInManager.SendTwoFactorCodeAsync(model.SelectedProvid
er))
    {
        return View("Error");
    }
    return RedirectToAction("VerifyCode", new {
Provider = model.SelectedProvider, ReturnUrl =
model.ReturnUrl, RememberMe = model.RememberMe });
}

// GET: /Account/ExternalLoginCallback
[AllowAnonymous]
public async Task<ActionResult>
ExternalLoginCallback(string returnUrl)
{
    var loginInfo = await
AuthenticationManager.GetExternalLoginInfoAsync();
    if (loginInfo == null)
    {
        return RedirectToAction("Login");
    }
}

```



```

        if (id == null)
        {
            return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Anggota anggota = db.Anggotas.Find(id);
        if (anggota == null)
        {
            return HttpNotFound();
        }
        ViewBag.Angg_Agama_FK = new
        SelectList(db.Agamas, "Agama_id", "Agama_nama",
        anggota.Angg_Agama_FK);
        ViewBag.Angg_Gender_FK = new
        SelectList(db.Genders, "Gend_id", "Gend_nama",
        anggota.Angg_Gender_FK);
        ViewBag.Angg_JenisIdentitas_FK = new
        SelectList(db.Identitas, "Iden_id", "Iden_nama",
        anggota.Angg_JenisIdentitas_FK);
        ViewBag.Angg_Kabupaten_FK = new
        SelectList(db.Kabupatens, "Kab_id", "Kab_nama",
        anggota.Angg_Kabupaten_FK);
        ViewBag.Angg_Status_FK = new
        SelectList(db.Statuses, "Status_id", "Status_nama",
        anggota.Angg_Status_FK);
        ViewBag.Anggota_Tahun_FK = new
        SelectList(db.Periodes, "Periode_id", "Periode_ke",
        anggota.Anggota_Tahun_FK);
        ViewBag.Angg_UserName = new
        SelectList(db.AspNetUsers.Where(t => t.AspNetRoles.Any(r
        => r.Name == "Nasabah")), "Id", "Email",
        anggota.Angg_UserName);
        return View(anggota);
    }

    // POST: Anggotas/Edit/5
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost, ActionName("Edit")]
    [ValidateAntiForgeryToken]
    public ActionResult EditPost(int? id,
    HttpPostedFileBase image1)
    {
        if (id == null)
        {
            return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        var anggotaToUpdate = db.Anggotas.Find(id);
        if (ModelState.IsValid)
        {
            if (image1 != null)
            {
                if (TryUpdateModel(anggotaToUpdate,
                "",
                new string[] { "Angg_Nama",
                "Angg_NomorAnggota", "Angg_Gender_FK", "Angg_Agama_FK",
                "Angg_Kabupaten_FK", "Angg_TanggalLahir",
                "Angg_JenisIdentitas_FK", "Angg_NomorIdentitas",
                "Angg_FotoIdentitas", "Anggota_Tahun_FK", "Angg_Alamat",
                "Angg_Status_FK", "Angg_UserName" }))
                {
                    try
                    {
                        anggotaToUpdate.Angg_FotoIdentitas = new
                        byte[image1.ContentLength];
                        image1.InputStream.Read(anggotaToUpdate.Angg_FotoIdentita
                        s, 0, image1.ContentLength);
                        db.Entry(anggotaToUpdate).State = EntityState.Modified;
                        db.SaveChanges();
                        return
                        RedirectToAction("Index");
                    }
                    catch
                    {
                        (RetryLimitExceededException)
                        {
                            ModelState.AddModelError("",
                            "Unable to save changes. Try again.");
                        }
                    }
                }
            }
            else
            {
                if (TryUpdateModel(anggotaToUpdate,
                "",

```

```

                new string[] { "Angg_Nama",
                "Angg_NomorAnggota", "Angg_Gender_FK", "Angg_Agama_FK",
                "Angg_Kabupaten_FK", "Angg_TanggalLahir",
                "Angg_JenisIdentitas_FK", "Angg_NomorIdentitas",
                "Anggota_Tahun_FK", "Angg_Alamat", "Angg_Status_FK",
                "Angg_UserName" }));
            {
                try
                {
                    db.Entry(anggotaToUpdate).State = EntityState.Modified;
                    db.SaveChanges();
                    return
                    RedirectToAction("Index");
                }
                catch
                {
                    (RetryLimitExceededException)
                    {
                        ModelState.AddModelError("",
                        "Unable to save changes. Try again.");
                    }
                }
            }
        }
        ViewBag.Angg_Agama_FK = new
        SelectList(db.Agamas, "Agama_id", "Agama_nama",
        anggotaToUpdate.Angg_Agama_FK);
        ViewBag.Angg_Gender_FK = new
        SelectList(db.Genders, "Gend_id", "Gend_nama",
        anggotaToUpdate.Angg_Gender_FK);
        ViewBag.Angg_JenisIdentitas_FK = new
        SelectList(db.Identitas, "Iden_id", "Iden_nama",
        anggotaToUpdate.Angg_JenisIdentitas_FK);
        ViewBag.Angg_Kabupaten_FK = new
        SelectList(db.Kabupatens, "Kab_id", "Kab_nama",
        anggotaToUpdate.Angg_Kabupaten_FK);
        ViewBag.Angg_Status_FK = new
        SelectList(db.Statuses, "Status_id", "Status_nama",
        anggotaToUpdate.Angg_Status_FK);
        ViewBag.Anggota_Tahun_FK = new
        SelectList(db.Periodes, "Periode_id", "Periode_ke",
        anggotaToUpdate.Anggota_Tahun_FK);
        ViewBag.Angg_UserName = new
        SelectList(db.AspNetUsers.Where(t => t.AspNetRoles.Any(r
        => r.Name == "Nasabah")), "Id", "Email",
        anggotaToUpdate.Angg_UserName);
        return View(anggotaToUpdate);
    }

    // GET: Anggotas/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Anggota anggota = db.Anggotas.Find(id);
        if (anggota == null)
        {
            return HttpNotFound();
        }
        return View(anggota);
    }

    // POST: Anggotas/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Anggota anggota = db.Anggotas.Find(id);
        db.Anggotas.Remove(anggota);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

3. Source Code Agama

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class AAgamasController : Controller
    {
        private BMT_IMEntities1 db = new BMT_IMEntities1();

        // GET: AAgamas
        public ActionResult Index()
        {
            return View(db.Agamas.ToList());
        }

        // GET: AAgamas/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Agama agama = db.Agamas.Find(id);
            if (agama == null)
            {
                return HttpNotFound();
            }
            return View(agama);
        }

        [Authorize(Roles = "Admin", Users =
        "SuperAdmin")]
        // GET: AAgamas/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AAgamas/Create
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Agama_id,Agama_nama")] Agama agama)
        {
            if (ModelState.IsValid)
            {
                db.Agamas.Add(agama);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(agama);
        }

        // GET: AAgamas/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Agama agama = db.Agamas.Find(id);
            if (agama == null)
            {
                return HttpNotFound();
            }
            return View(agama);
        }

        // POST: AAgamas/Edit/5
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
        "Agama_id,Agama_nama")] Agama agama)
        {
            if (ModelState.IsValid)
            {
                db.Entry(agama).State =
                EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(agama);
        }

        // GET: AAgamas/Delete/5
        public ActionResult Delete(int? id)
        {
            if (id == null)
            {

```

```

                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Agama agama = db.Agamas.Find(id);
            if (agama == null)
            {
                return HttpNotFound();
            }
            return View(agama);
        }

        // POST: AAgamas/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Agama agama = db.Agamas.Find(id);
            db.Agamas.Remove(agama);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}

```

4. Source Code Gender

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class AGendersController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        // GET: AGenders
        public ActionResult Index()
        {
            return View(db.Genders.ToList());
        }

        // GET: AGenders/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Gender gender = db.Genders.Find(id);
            if (gender == null)
            {
                return HttpNotFound();
            }
            return View(gender);
        }

        [Authorize(Roles = "Admin", Users =
        "SuperAdmin")]
        // GET: AGenders/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AGenders/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        // http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Gend_id,Gend_nama")] Gender gender)
        {
            if (ModelState.IsValid)
            {
                db.Genders.Add(gender);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
        }
    }
}

```

```

        return View(gender);
    }

    // GET: AGenders/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Gender gender = db.Genders.Find(id);
        if (gender == null)
        {
            return HttpNotFound();
        }
        return View(gender);
    }

    // POST: AGenders/Edit/5
    // To protect from overposting attacks, please
    // enable the specific properties you want to bind to, for
    // more details see
    // http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "Gend_id,Gend_nama")] Gender gender)
    {
        if (ModelState.IsValid)
        {
            db.Entry(gender).State =
            EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(gender);
    }

    // GET: AGenders/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Gender gender = db.Genders.Find(id);
        if (gender == null)
        {
            return HttpNotFound();
        }
        return View(gender);
    }

    // POST: AGenders/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Gender gender = db.Genders.Find(id);
        db.Genders.Remove(gender);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

5. Source Code Identitas

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class AIdentitasController : Controller
    {
        private BMT_IMEntities1 db = new BMT_IMEntities1();

        // GET: AIdentitas
        public ActionResult Index()
        {

```

```

            return View(db.Identitas.ToList());
        }

        // GET: AIdentitas/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Identita identita = db.Identitas.Find(id);
            if (identita == null)
            {
                return HttpNotFound();
            }
            return View(identita);
        }

        [Authorize(Roles = "Admin", Users = "SuperAdmin")]
        // GET: AIdentitas/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AIdentitas/Create
        // To protect from overposting attacks, please enable the
        // specific properties you want to bind to, for
        // more details see
        // http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Iden_id,Iden_nama")] Identita identita)
        {
            if (ModelState.IsValid)
            {
                db.Identitas.Add(identita);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(identita);
        }

        // GET: AIdentitas/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Identita identita = db.Identitas.Find(id);
            if (identita == null)
            {
                return HttpNotFound();
            }
            return View(identita);
        }

        // POST: AIdentitas/Edit/5
        // To protect from overposting attacks, please enable the
        // specific properties you want to bind to, for
        // more details see
        // http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
        "Iden_id,Iden_nama")] Identita identita)
        {
            if (ModelState.IsValid)
            {
                db.Entry(identita).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(identita);
        }

        // GET: AIdentitas/Delete/5
        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Identita identita = db.Identitas.Find(id);
            if (identita == null)

```

```

{
return HttpNotFound();
}
return View(Identita);
}

// POST: AIdentitas/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
Identita identita = db.Identitas.Find(id);
db.Identitas.Remove(Identita);
db.SaveChanges();
return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
if (disposing)
{
db.Dispose();
}
base.Dispose(disposing);
}
}

```

6. Source Code Kabupaten

```

namespace SIBMT_INSANMADANI.Controllers
{
public class AKabupatensController : Controller
{
private BMT_IMEntities1 db = new
BMT_IMEntities1();

// GET: AKabupatens
public ActionResult Index()
{
var kabupatens = db.Kabupatens.Include(k =>
k.Provinsi);
return View(kabupatens.ToList());
}

// GET: AKabupatens/Details/5
public ActionResult Details(int? id)
{
if (id == null)
{
return
HttpStatusCodeResult(HttpStatusCode.BadRequest);
}
Kabupaten kabupaten =
db.Kabupatens.Find(id);
if (kabupaten == null)
{
return HttpNotFound();
}
return View(kabupaten);
}

[Authorize(Roles = "Admin", Users =
"SuperAdmin")]
// GET: AKabupatens/Create
public ActionResult Create()
{
ViewBag.Prov_id_FK =
SelectList(db.Provinsis, "Prov_id", "Prov_nama");
return View();
}

// POST: AKabupatens/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"Kab_id,Prov_id_FK,Kab_nama")] Kabupaten kabupaten)
{
if (ModelState.IsValid)
{
db.Kabupatens.Add(kabupaten);
db.SaveChanges();
return RedirectToAction("Index");
}

ViewBag.Prov_id_FK =
SelectList(db.Provinsis, "Prov_id", "Prov_nama",
kabupaten.Prov_id_FK);
}
}

```

```

return View(kabupaten);
}

// GET: AKabupatens/Edit/5
public ActionResult Edit(int? id)
{
if (id == null)
{
return
HttpStatusCodeResult(HttpStatusCode.BadRequest);
}
Kabupaten kabupaten =
db.Kabupatens.Find(id);
if (kabupaten == null)
{
return HttpNotFound();
}
ViewBag.Prov_id_FK =
SelectList(db.Provinsis, "Prov_id", "Prov_nama",
kabupaten.Prov_id_FK);
return View(kabupaten);
}

// POST: AKabupatens/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Kab_id,Prov_id_FK,Kab_nama")] Kabupaten kabupaten)
{
if (ModelState.IsValid)
{
db.Entry(kabupaten).State =
EntityState.Modified;
db.SaveChanges();
return RedirectToAction("Index");
}
ViewBag.Prov_id_FK =
SelectList(db.Provinsis, "Prov_id", "Prov_nama",
kabupaten.Prov_id_FK);
return View(kabupaten);
}

// GET: AKabupatens/Delete/5
public ActionResult Delete(int? id)
{
if (id == null)
{
return
HttpStatusCodeResult(HttpStatusCode.BadRequest);
}
Kabupaten kabupaten =
db.Kabupatens.Find(id);
if (kabupaten == null)
{
return HttpNotFound();
}
return View(kabupaten);
}

// POST: AKabupatens/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
Kabupaten kabupaten =
db.Kabupatens.Find(id);
db.Kabupatens.Remove(kabupaten);
db.SaveChanges();
return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
if (disposing)
{
db.Dispose();
}
base.Dispose(disposing);
}
}
}

```


7. Source Code Provinsi

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class AProvinsiController : Controller
    {
        private BMT_IMEntities db = new
        BMT_IMEntities();

        // GET: AProvinsi
        public ActionResult Index()
        {
            return View(db.Provinsis.ToList());
        }

        // GET: AProvinsi/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest); new
            }
            Provinsi provinsi = db.Provinsis.Find(id);
            if (provinsi == null)
            {
                return HttpNotFound();
            }
            return View(provinsi);
        }

        [Authorize(Roles = "Admin", Users =
        "SuperAdmin")]
        // GET: AProvinsi/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AProvinsi/Create
        // To protect from overposting attacks, please
        enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Prov_id,Prov_nama")] Provinsi provinsi)
        {
            if (ModelState.IsValid)
            {
                db.Provinsis.Add(provinsi);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(provinsi);
        }

        // GET: AProvinsi/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest); new
            }
            Provinsi provinsi = db.Provinsis.Find(id);
            if (provinsi == null)
            {
                return HttpNotFound();
            }
            return View(provinsi);
        }

        // POST: AProvinsi/Edit/5
        // To protect from overposting attacks, please
        enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
        "Prov_id,Prov_nama")] Provinsi provinsi)
        {
            if (ModelState.IsValid)
            {
                db.Entry(provinsi).State
                EntityState.Modified;
                db.SaveChanges();
            }
        }
    }
}

```

```

        return RedirectToAction("Index");
    }
    return View(provinsi);
}

// GET: AProvinsi/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return
        HttpStatusCodeResult(HttpStatusCode.BadRequest); new
    }
    Provinsi provinsi = db.Provinsis.Find(id);
    if (provinsi == null)
    {
        return HttpNotFound();
    }
    return View(provinsi);
}

// POST: AProvinsi/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Provinsi provinsi = db.Provinsis.Find(id);
    db.Provinsis.Remove(provinsi);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

8. Source Code Periode

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class AProvinsiController : Controller
    {
        private BMT_IMEntities db = new
        BMT_IMEntities();

        // GET: AProvinsi
        public ActionResult Index()
        {
            return View(db.Provinsis.ToList());
        }

        // GET: AProvinsi/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest); new
            }
            Provinsi provinsi = db.Provinsis.Find(id);
            if (provinsi == null)
            {
                return HttpNotFound();
            }
            return View(provinsi);
        }

        [Authorize(Roles = "Admin", Users =
        "SuperAdmin")]
        // GET: AProvinsi/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AProvinsi/Create
        // To protect from overposting attacks, please
        enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
    }
}

```

```

        public ActionResult Create([Bind(Include =
"Prov_id,Prov_nama")] Provinsi provinsi)
        {
            if (ModelState.IsValid)
            {
                db.Provinsis.Add(provinsi);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(provinsi);
        }

        // GET: AProvinsis/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Provinsi provinsi = db.Provinsis.Find(id);
            if (provinsi == null)
            {
                return HttpNotFound();
            }
            return View(provinsi);
        }

        // POST: AProvinsis/Edit/5
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
"Prov_id,Prov_nama")] Provinsi provinsi)
        {
            if (ModelState.IsValid)
            {
                db.Entry(provinsi).State =
                EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(provinsi);
        }

        // GET: AProvinsis/Delete/5
        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Provinsi provinsi = db.Provinsis.Find(id);
            if (provinsi == null)
            {
                return HttpNotFound();
            }
            return View(provinsi);
        }

        // POST: AProvinsis/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Provinsi provinsi = db.Provinsis.Find(id);
            db.Provinsis.Remove(provinsi);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}

```

9. Source Code Status

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class AStatusesController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        // GET: AStatuses
        public ActionResult Index()
        {
            return View(db.Statuses.ToList());
        }

        // GET: AStatuses/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Status status = db.Statuses.Find(id);
            if (status == null)
            {
                return HttpNotFound();
            }
            return View(status);
        }

        [Authorize(Roles = "Admin", Users =
"SuperAdmin")]
        // GET: AStatuses/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: AStatuses/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"Status_id,Status_nama")] Status status)
        {
            if (ModelState.IsValid)
            {
                db.Statuses.Add(status);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(status);
        }

        // GET: AStatuses/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Status status = db.Statuses.Find(id);
            if (status == null)
            {
                return HttpNotFound();
            }
            return View(status);
        }

        // POST: AStatuses/Edit/5
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit([Bind(Include =
"Status_id,Status_nama")] Status status)
        {
            if (ModelState.IsValid)
            {
                db.Entry(status).State =
                EntityState.Modified;
                db.SaveChanges();
            }
        }
    }
}

```

```

        return RedirectToAction("Index");
    }
    return View(status);
}

// GET: AStatuses/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Status status = db.Statuses.Find(id);
    if (status == null)
    {
        return HttpNotFound();
    }
    return View(status);
}

// POST: AStatuses/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Status status = db.Statuses.Find(id);
    db.Statuses.Remove(status);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

10. Source Code Home

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class HomeController : Controller
    {
        BMT_IMEntities1 db = new BMT_IMEntities1();
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult Report()
        {
            string tanggal = DateTime.Now.ToString("MM-
            dd-yyyy");
            //string tanggal = "07-14-2017";
            ReportDocument rd = new ReportDocument();

            rd.Load(Path.Combine(Server.MapPath("~/Report/Laporan.r
            pt")));
            rd.SetDataSource(db.Report_Harian.Select(p
            => new
            {
                Tanggal = p.Tanggal,
                Kode = p.Kode,
                Nasabah = p.Nasabah,
                Pinjam_NoRek = p.Pinjam_NoRek,
                Debet = p.Debet,
                Kredit = p.Kredit,
                Penerima = p.Penerima,
                Ke = p.Ke
            }).Where(p => p.Tanggal ==
            tanggal).OrderBy(p => p.Nasabah).ToList());
            //var c = (from b in db.Report_Harian select
            b).Where(b => b.Tanggal == tanggal).OrderBy(b =>
            b.Nasabah).ToList();
            //Laporan rpt = new Laporan();
            //rpt.Load();
            //rpt.SetDataSource(c);
            //Stream stream =
            rpt.ExportToStream(CrystalDecisions.Shared.ExportFormat
            Type.PortableDocFormat);

            Response.Buffer = false;

```

```

        Response.ClearContent();
        Response.ClearHeaders();
        Stream stream = rd.ExportToStream

(CrystalDecisions.Shared.ExportFormatType.PortableDocFo
rmat);
        stream.Seek(0, SeekOrigin.Begin);
        return File(stream, "application/pdf");
    }
    [Authorize]
    public ActionResult Simpan()
    {
        ViewBag.Message = "Your application
        description page.";

        return View();
    }
    [Authorize]
    public ActionResult Infaq()
    {
        ViewBag.Message = "Your application
        description page.";

        return View();
    }
    [Authorize]
    public ActionResult Pinjam()
    {
        ViewBag.Message = "Your contact page.";

        return View();
    }
}
}

```

11. Source Code Nominal_P

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Nominal_PController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        // GET: Nominal_P
        public ActionResult Index()
        {
            return View(db.Nominal_P.ToList());
        }

        // GET: Nominal_P/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Nominal_P nominal_P = db.Nominal_P.Find(id);
            if (nominal_P == null)
            {
                return HttpNotFound();
            }
            return View(nominal_P);
        }

        // GET: Nominal_P/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Nominal_P/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Nominal_id,Nominal_Terbilang")] Nominal_P nominal_P)
        {
            if (ModelState.IsValid)
            {
                db.Nominal_P.Add(nominal_P);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

```

```

    }
    return View(nominal_P);
}

// GET: Nominal_P/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Nominal_P nominal_P = db.Nominal_P.Find(id);
    if (nominal_P == null)
    {
        return HttpNotFound();
    }
    return View(nominal_P);
}

// POST: Nominal_P/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Nominal_id,Nominal_Terbilang")] Nominal_P nominal_P)
{
    if (ModelState.IsValid)
    {
        db.Entry(nominal_P).State =
        EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(nominal_P);
}

// GET: Nominal_P/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Nominal_P nominal_P = db.Nominal_P.Find(id);
    if (nominal_P == null)
    {
        return HttpNotFound();
    }
    return View(nominal_P);
}

// POST: Nominal_P/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Nominal_P nominal_P = db.Nominal_P.Find(id);
    db.Nominal_P.Remove(nominal_P);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

12. Source Code Nominal_SP

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Nominal_SPController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
    }
}

```

```

// GET: Nominal_SP
public ActionResult Index()
{
    return View(db.Nominal_SP.ToList());
}

// GET: Nominal_SP/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Nominal_SP nominal_SP =
    db.Nominal_SP.Find(id);
    if (nominal_SP == null)
    {
        return HttpNotFound();
    }
    return View(nominal_SP);
}

// GET: Nominal_SP/Create
public ActionResult Create()
{
    return View();
}

// POST: Nominal_SP/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"Nominal_id,Nominal_Terbilang")] Nominal_SP nominal_SP)
{
    if (ModelState.IsValid)
    {
        db.Nominal_SP.Add(nominal_SP);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(nominal_SP);
}

// GET: Nominal_SP/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Nominal_SP nominal_SP =
    db.Nominal_SP.Find(id);
    if (nominal_SP == null)
    {
        return HttpNotFound();
    }
    return View(nominal_SP);
}

// POST: Nominal_SP/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Nominal_id,Nominal_Terbilang")] Nominal_SP nominal_SP)
{
    if (ModelState.IsValid)
    {
        db.Entry(nominal_SP).State =
        EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(nominal_SP);
}

// GET: Nominal_SP/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {

```

```

        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Nominal_SP nominal_SP =
    db.Nominal_SP.Find(id);
    if (nominal_SP == null)
    {
        return HttpNotFound();
    }
    return View(nominal_SP);

    // POST: Nominal_SP/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Nominal_SP nominal_SP =
        db.Nominal_SP.Find(id);
        db.Nominal_SP.Remove(nominal_SP);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

13. Source Code Nominal_SW

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Nominal_SWController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        // GET: Nominal_SW
        public ActionResult Index()
        {
            return View(db.Nominal_SW.ToList());
        }

        // GET: Nominal_SW/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Nominal_SW nominal_SW =
            db.Nominal_SW.Find(id);
            if (nominal_SW == null)
            {
                return HttpNotFound();
            }
            return View(nominal_SW);
        }

        // GET: Nominal_SW/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Nominal_SW/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Nominal_id,Nominal_Terbilang")] Nominal_SW nominal_SW)
        {
            if (ModelState.IsValid)
            {
                db.Nominal_SW.Add(nominal_SW);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
        }
    }
}

```

```

        return View(nominal_SW);
    }

    // GET: Nominal_SW/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Nominal_SW nominal_SW =
        db.Nominal_SW.Find(id);
        if (nominal_SW == null)
        {
            return HttpNotFound();
        }
        return View(nominal_SW);
    }

    // POST: Nominal_SW/Edit/5
    // To protect from overposting attacks, please
    // enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "Nominal_id,Nominal_Terbilang")] Nominal_SW nominal_SW)
    {
        if (ModelState.IsValid)
        {
            db.Entry(nominal_SW).State =
            EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(nominal_SW);
    }

    // GET: Nominal_SW/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Nominal_SW nominal_SW =
        db.Nominal_SW.Find(id);
        if (nominal_SW == null)
        {
            return HttpNotFound();
        }
        return View(nominal_SW);
    }

    // POST: Nominal_SW/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Nominal_SW nominal_SW =
        db.Nominal_SW.Find(id);
        db.Nominal_SW.Remove(nominal_SW);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

14. Source Code Pengelola

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class PengelolasController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        // GET: Pengelolas
        public ActionResult Index()
        {
            //var result = db.AspNetUsers.Where(t =>
            t.AspNetRoles.Any(r => r.Name == "Admin")).Select(x =>
            x).ToList();
            var pengelolas = db.Pengelolas.Include(p =>
            p.AspNetUser);
            return View(pengelolas.ToList());
            //return View(pengelolas.ToList());
        }

        // GET: Pengelolas/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Pengelola pengelola =
            db.Pengelolas.Find(id);
            if (pengelola == null)
            {
                return HttpNotFound();
            }
            return View(pengelola);
        }

        // GET: Pengelolas/Create
        [Authorize(Roles = "Admin", Users =
        "SuperAdmin")]
        public ActionResult Create(string id)
        {
            //var result = db.AspNetUsers.Where(t =>
            t.AspNetRoles.Any(r => r.Name == "Admin")).Select(x =>
            x).ToList();

            ViewBag.Pengelola_UserName = new
            SelectList(db.AspNetUsers.Where(t => t.Id == id), "id",
            "Email");
            return View();
        }

        // POST: Pengelolas/Create
        // To protect from overposting attacks, please
        enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "Pengelola_id,Pengelola_Nama,Pengelola_UserName")]
        Pengelola pengelola)
        {
            if (ModelState.IsValid)
            {
                db.Pengelolas.Add(pengelola);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
        }

        ViewBag.Pengelola_UserName = new
        SelectList(db.AspNetUsers, "Id", "Email",
        pengelola.Pengelola_UserName);
        return View(pengelola);
    }

    // GET: Pengelolas/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Pengelola pengelola =
        db.Pengelolas.Find(id);
        if (pengelola == null)
        {
            return HttpNotFound();
        }
    }
}

```

```

        ViewBag.Pengelola_UserName = new
        SelectList(db.AspNetUsers, "Id", "Email",
        pengelola.Pengelola_UserName);
        return View(pengelola);
    }

    // POST: Pengelolas/Edit/5
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "Pengelola_id,Pengelola_Nama,Pengelola_UserName")]
    Pengelola pengelola)
    {
        if (ModelState.IsValid)
        {
            db.Entry(pengelola).State
            EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.Pengelola_UserName = new
        SelectList(db.AspNetUsers, "Id", "Email",
        pengelola.Pengelola_UserName);
        return View(pengelola);
    }

    // GET: Pengelolas/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Pengelola pengelola =
        db.Pengelolas.Find(id);
        if (pengelola == null)
        {
            return HttpNotFound();
        }
        return View(pengelola);
    }

    // POST: Pengelolas/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Pengelola pengelola =
        db.Pengelolas.Find(id);
        db.Pengelolas.Remove(pengelola);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

15. Source Code Trans_SimpananPokok

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_SimpanPokokController :
    Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
        private Random random = new Random();
        string connectionString =
        ConfigurationManager.ConnectionStrings["DefaultConnecti
        on"].ConnectionString;
        SqlConnection con;
        SqlCommand com;
        void Rupiah()
        {
        }
    }
}

```

```

System.Threading.Thread.CurrentThread.CurrentCulture =
new System.Globalization.CultureInfo("id-ID");

System.Threading.Thread.CurrentThread.CurrentUICulture =
new System.Globalization.CultureInfo("id-ID");
}
// GET: Trans_SimpanPokok
public ActionResult Index()
{
    Rupiah();
    var trans_SimpanPokok =
db.Trans_SimpanPokok.Include(t => t.Anggota).Include(t
=> t.Pengelola).Include(t => t.Nominal_SP);
    return View(trans_SimpanPokok.ToList());
}
public ActionResult Export(int? id)
{
    ReportDocument rd = new ReportDocument();

rd.Load(Path.Combine(Server.MapPath("~/Report/Sim_Pokok
.rpt")));

rd.SetDataSource(db.Trans_SimpanPokok.Select(p => new
{
    SimpanP_id = p.SimpanP_id,
    SimpanP_Kode_FK =
p.Trans_Kode.Kode_Transaksi,
    SimpanP_NamaAnggota_FK =
p.Anggota.Angg_Nama,
    SimpanP_NoRek = p.SimpanP_NoRek,
    SimpanP_Nominal_FK =
p.Nominal_SP.Nominal_Terbilang,
    SimpanP_Tanggal = p.SimpanP_Tanggal,
    SimpanP_Penerima_FK =
p.Pengelola.Pengelola_Nama,
    SimpanP_Saldo = p.SimpanP_Saldo

}).Where(p => p.SimpanP_id == id).ToList());
Response.Buffer = false;
Response.ClearContent();
Response.ClearHeaders();
Stream stream = rd.ExportToStream

(CrystalDecisions.Shared.ExportFormatType.PortableDocFo
rmat);
    stream.Seek(0, SeekOrigin.Begin);
    return File(stream, "application/pdf");
}
// GET: Trans_SimpanPokok/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    Trans_SimpanPokok trans_SimpanPokok =
db.Trans_SimpanPokok.Find(id);
    if (trans_SimpanPokok == null)
    {
        return HttpNotFound();
    }
    return View(trans_SimpanPokok);
}

[Authorize]
public ActionResult CreateAwal(int? status)
{
    con = new SqlConnection(connectionString);
    string id = "";
    string userName = "";
    string pesan = "";
    try
    {
        con.Open();
        com = new SqlCommand("Select id from
dbo.AspNetUsers where UserName = '" + User.Identity.Name
+ "'", con);
        id = (string)com.ExecuteScalar();
        com = new SqlCommand("Select
Pengelola_Nama from Pengelola where Pengelola_UserName =
'" + id + "'", con);
        userName = (string)com.ExecuteScalar();
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
}

```

```

ViewBag.SimpanP_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_Id", "Kode_Transaksi");
ViewBag.SimpanP_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
ViewBag.SimpanP_Nominal_FK = new
SelectList(db.Nominal_SP, "Nominal_id",
"Nominal_Terbilang");
ViewBag.SimpanP_Penerima_FK = userName;
if (status != null)
{
    pesan = "Anda Sudah Melakukan Simpanan
Pokok. Simpanan Pokok hanya 1x !";
}
ViewBag.Status = pesan;
return View();
}
// GET: Trans_SimpanPokok/Create
[HttpGet]
public ActionResult Create(String kode, String
nama, int nominal, DateTime tanggal, String penerima)
{
    string noRekDb = "";
    string satu = "";
    int no = 0;
    int IDsimpanPokok = 0;
    int ID = Convert.ToInt32(nama);
    double jumlah;
    con = new
SqlConnection(connectionString);
    //mencari id Simpanan Pokok
    try
    {
        con.Open();
        com = new SqlCommand("select
SimpanP_id from dbo.Trans_SimpanPokok where
SimpanP_NamaAnggota_FK= " + ID, con);
        IDsimpanPokok =
Convert.ToInt32(com.ExecuteScalar());
        com = new SqlCommand("select
Nominal_Terbilang from dbo.Nominal_SP where Nominal_id=
" + nominal, con);
        jumlah =
Convert.ToDouble(com.ExecuteScalar());
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
    //membuat nomor rekening
    Anggota nasa = db.Anggotas.Find(ID);
    string tgl =
nasa.Angg_TanggalLahir.ToString("yyyyMMdd");
    if (IDsimpanPokok != 0)//jika simpan
pokok sudah memiliki data
    {
        return
RedirectToAction("CreateAwal", new { status = 1 });
    }
    //return
RedirectToAction("CreateAwal");
}
else//jika simpan pokok belum memiliki
data
{
    no = random.Next(10000, 99999);
    satu = tgl + no.ToString();

    con.Open();
    com = new SqlCommand("select
SimpanP_NoRek from dbo.Trans_SimpanPokok where
SimpanP_NoRek = " + satu, con);
    noRekDb =
(string)com.ExecuteScalar();
    con.Close();

    if (satu == noRekDb)//jika nomor
rekeningnya sudah digunakan
    {
        do
        {
            no = random.Next(10000,
99999);
            satu = tgl + no.ToString();
        } while (satu == noRekDb);
    }
    //mencari id Pengelola
    try
    {
        con.Open();
    }
}

```

```

        com = new SqlCommand("Select
Pengelola_id from Pengelola where Pengelola_Nama = '" +
penerima + "'", con);
        int idPengelola =
Convert.ToInt32(com.ExecuteScalar());
        con.Close();
        ViewBag.Penerima = idPengelola;
    }
    catch (Exception x)
    {
        throw x;
    }
    ViewBag.Kode = kode;
    ViewBag>Nama = nama;
    ViewBag.Nominal = nominal;
    ViewBag.Tanggal =
tanggal.ToString("yyyy-MM-dd");
    ViewBag.NoRek = satu;
    ViewBag.Saldo = jumlah;
    return View();
}

// POST: Trans_SimpanPokok/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"SimpanP_id,SimpanP_Kode_FK,SimpanP>NamaAnggota_FK,Simp
anP_NoRek,SimpanP_Nominal_FK,SimpanP_Tanggal,SimpanP_Pe
rnerima_FK,SimpanP_Saldo")] String kode, String nama,
double nominal, Trans_SimpanPokok trans_SimpanPokok)
{
    if (ModelState.IsValid)
    {
        try
        {
            con = new
SqlConnection(connectionString);
            con.Open();
            com = new SqlCommand("select
Nominal_Terbilang from dbo.Nominal_SP where Nominal_id=
" + nominal, con);
            double jumlah =
Convert.ToDouble(com.ExecuteScalar());
            com = new SqlCommand("insert into
dbo.SaldoSimpanan values(" + nama + ", " + jumlah + ")",
con);
            com.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception x)
        {
            throw x;
        }
    }

    db.Trans_SimpanPokok.Add(trans_SimpanPokok);
    db.SaveChanges();
    return RedirectToAction("Details", new {
id = trans_SimpanPokok.SimpanP_id });
}

ViewBag.SimpanP_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanPokok.SimpanP_Kode_FK);
ViewBag.SimpanP>NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg>Nama",
trans_SimpanPokok.SimpanP>NamaAnggota_FK);
ViewBag.SimpanP_Nominal_FK = new
SelectList(db.Nominal_SP, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanPokok.SimpanP_Nominal_FK);
ViewBag.SimpanP_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola>Nama",
trans_SimpanPokok.SimpanP_Pernerima_FK);
return View(trans_SimpanPokok);
}

// GET: Trans_SimpanPokok/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return
HttpStatusCodesResult(HttpStatusCode.BadRequest);
    }
    Trans_SimpanPokok trans_SimpanPokok =
db.Trans_SimpanPokok.Find(id);
    if (trans_SimpanPokok == null)
    {
        return
HttpStatusCodesResult(HttpStatusCode.BadRequest);
    }
    ViewBag.SimpanP_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanPokok.SimpanP_Kode_FK);
ViewBag.SimpanP>NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg>Nama",
trans_SimpanPokok.SimpanP>NamaAnggota_FK);
ViewBag.SimpanP_Nominal_FK = new
SelectList(db.Nominal_SP, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanPokok.SimpanP_Nominal_FK);
ViewBag.SimpanP_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola>Nama",
trans_SimpanPokok.SimpanP_Pernerima_FK);
return View(trans_SimpanPokok);
}

// POST: Trans_SimpanPokok/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"SimpanP_id,SimpanP_Kode_FK,SimpanP>NamaAnggota_FK,Simp
anP_NoRek,SimpanP_Nominal_FK,SimpanP_Tanggal,SimpanP_Pe
rnerima_FK,SimpanP_Saldo")] Trans_SimpanPokok
trans_SimpanPokok)
{
    if (ModelState.IsValid)
    {
        try
        {
            con = new
SqlConnection(connectionString);
            con.Open();
            com = new SqlCommand("select
Nominal_Terbilang from dbo.Nominal_SP where Nominal_id=
" + nominal, con);
            double jumlah =
Convert.ToDouble(com.ExecuteScalar());
            com = new SqlCommand("insert into
dbo.SaldoSimpanan values(" + nama + ", " + jumlah + ")",
con);
            com.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception x)
        {
            throw x;
        }
    }

    db.Trans_SimpanPokok.Add(trans_SimpanPokok);
    db.SaveChanges();
    return RedirectToAction("Details", new {
id = trans_SimpanPokok.SimpanP_id });
}

ViewBag.SimpanP_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanPokok.SimpanP_Kode_FK);
ViewBag.SimpanP>NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg>Nama",
trans_SimpanPokok.SimpanP>NamaAnggota_FK);
ViewBag.SimpanP_Nominal_FK = new
SelectList(db.Nominal_SP, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanPokok.SimpanP_Nominal_FK);
ViewBag.SimpanP_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola>Nama",
trans_SimpanPokok.SimpanP_Pernerima_FK);
return View(trans_SimpanPokok);
}

// GET: Trans_SimpanPokok/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return
HttpStatusCodesResult(HttpStatusCode.BadRequest);
    }
    Trans_SimpanPokok trans_SimpanPokok =
db.Trans_SimpanPokok.Find(id);
    if (trans_SimpanPokok == null)
    {
        return
HttpStatusCodesResult(HttpStatusCode.BadRequest);
    }
    db.Trans_SimpanPokok.Remove(trans_SimpanPokok);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)

```

```

    {
        return HttpNotFound();
    }
    ViewBag.SimpanP_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanPokok.SimpanP_Kode_FK);
    ViewBag.SimpanP>NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg>Nama",
trans_SimpanPokok.SimpanP>NamaAnggota_FK);
    ViewBag.SimpanP_Nominal_FK = new
SelectList(db.Nominal_SP, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanPokok.SimpanP_Nominal_FK);
    ViewBag.SimpanP_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola>Nama",
trans_SimpanPokok.SimpanP_Pernerima_FK);
    return View(trans_SimpanPokok);
}

// POST: Trans_SimpanPokok/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"SimpanP_id,SimpanP_Kode_FK,SimpanP>NamaAnggota_FK,Simp
anP_NoRek,SimpanP_Nominal_FK,SimpanP_Tanggal,SimpanP_Pe
rnerima_FK,SimpanP_Saldo")] Trans_SimpanPokok
trans_SimpanPokok)
{
    if (ModelState.IsValid)
    {
        db.Entry(trans_SimpanPokok).State =
EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.SimpanP_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanPokok.SimpanP_Kode_FK);
    ViewBag.SimpanP>NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg>Nama",
trans_SimpanPokok.SimpanP>NamaAnggota_FK);
    ViewBag.SimpanP_Nominal_FK = new
SelectList(db.Nominal_SP, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanPokok.SimpanP_Nominal_FK);
    ViewBag.SimpanP_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola>Nama",
trans_SimpanPokok.SimpanP_Pernerima_FK);
    return View(trans_SimpanPokok);
}

// GET: Trans_SimpanPokok/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return
HttpStatusCodesResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    Trans_SimpanPokok trans_SimpanPokok =
db.Trans_SimpanPokok.Find(id);
    if (trans_SimpanPokok == null)
    {
        return
HttpStatusCodesResult(HttpStatusCode.BadRequest);
    }
    return View(trans_SimpanPokok);
}

// POST: Trans_SimpanPokok/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Trans_SimpanPokok trans_SimpanPokok =
db.Trans_SimpanPokok.Find(id);
    db.Trans_SimpanPokok.Remove(trans_SimpanPokok);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)

```



```

        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

16. Source Code Trans_SimpananWajib

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_SimpanWajibController :
    Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
        private Random random = new Random();
        string connectionString =
        ConfigurationManager.ConnectionStrings["DefaultConnecti
        on"].ConnectionString;
        SqlConnection con;
        SqlCommand com;

        void Rupiah()
        {
            System.Threading.Thread.CurrentThread.CurrentCulture =
            new System.Globalization.CultureInfo("id-ID");

            System.Threading.Thread.CurrentThread.CurrentUICulture =
            new System.Globalization.CultureInfo("id-ID");
        }
        // GET: Trans_SimpanWajib
        public ActionResult Index()
        {
            Rupiah();
            var trans_SimpanWajib =
            db.Trans_SimpanWajib.Include(t => t.Anggota).Include(t
            => t.Pengelola).Include(t => t.Nominal_SW);
            return View(trans_SimpanWajib.ToList());
        }
        public ActionResult Export(int? id)
        {
            ReportDocument rd = new ReportDocument();

            rd.Load(Path.Combine(Server.MapPath("~/report/Sim_Wajib
            .rpt")));

            rd.SetDataSource(db.Trans_SimpanWajib.Select(p => new
            {
                SimpanW_id = p.SimpanW_id,
                SimpanW_Kode_FK =
                p.Trans_Kode.Kode_Transaksi,
                SimpanW_NamaAnggota_FK =
                p.Anggota.Angg_Nama,
                SimpanW_NoRek = p.SimpanW_NoRek,
                SimpanW_Nominal_FK =
                p.Nominal_SW.Nominal_Terbilang,
                SimpanW_Tanggal = p.SimpanW_Tanggal,
                SimpanW_Penerima_FK =
                p.Pengelola.Pengelola_Nama,
                SimpanW_Saldo = p.SimpanW_Saldo
            }).Where(p => p.SimpanW_id == id).ToList());
            Response.Buffer = false;
            Response.ClearContent();
            Response.ClearHeaders();
            Stream stream = rd.ExportToStream

            (CrystalDecisions.Shared.ExportFormatType.PortableDocFo
            rmat);

            stream.Seek(0, SeekOrigin.Begin);
            return File(stream, "application/pdf");
        }
        // GET: Trans_SimpanWajib/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Rupiah();
            Trans_SimpanWajib trans_SimpanWajib =
            db.Trans_SimpanWajib.Find(id);
            if (trans_SimpanWajib == null)
            {
                return HttpNotFound();
            }
        }
    }
}

```

```

    }
    return View(trans_SimpanWajib);
}

[Authorize]
public ActionResult CreateAwal(int? status)
{
    con = new SqlConnection(connectionString);
    string id = "";
    string userName = "";
    string pesan = "";
    //mendapatkan pengelola
    try
    {
        con.Open();
        com = new SqlCommand("Select id from
        dbo.AspNetUsers where UserName = '" + User.Identity.Name
        + "'", con);
        id = (string)com.ExecuteScalar();
        com = new SqlCommand("Select
        Pengelola_Nama from Pengelola where Pengelola_UserName =
        '" + id + "'", con);
        userName = (string)com.ExecuteScalar();
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
    ViewBag.SimpanW_Kode_FK =
    new SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi");
    ViewBag.SimpanW_NamaAnggota_FK =
    new SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
    ViewBag.SimpanW_Nominal_FK =
    new SelectList(db.Nominal_SW, "Nominal_id",
    "Nominal_Terbilang");
    ViewBag.SimpanW_Penerima_FK = userName;
    if (status != null)
    {
        pesan = "Anda belum melakukan simpanan
        pokok !";
    }
    ViewBag.Status = pesan;
    return View();
}
[HttpGet]
// GET: Trans_SimpanWajib/Create
public ActionResult Create(String kode, String
nama, int nominal, DateTime tanggal, String penerima)
{
    string noRekDb = "";
    double saldo = 0;
    int IDsimpanPokok = 0;
    int IDsimpanWajib = 0;
    double jumlah;
    string status = "";
    DateTime TglSP;
    DateTime TglHarusSimpan;
    DateTime TglISWNext;
    DateTime TglISW = tanggal;

    int ID = Convert.ToInt32(nama);
    con = new SqlConnection(connectionString);
    //mendapatkan id nasabah dan jumlah saldo
    yang dimiliki
    try
    {
        con.Open();
        com = new SqlCommand("select SimpanP_id
        from dbo.Trans_SimpanPokok where SimpanP_NamaAnggota_FK=
        " + ID, con);
        IDsimpanPokok =
        Convert.ToInt32(com.ExecuteScalar());
        com = new SqlCommand("select
        SimpanP_Tanggal from Trans_SimpanPokok where
        SimpanP_NamaAnggota_FK = " + ID, con);
        TglSP =
        Convert.ToDateTime(com.ExecuteScalar());
        com = new SqlCommand("select
        Saldo_Sumlah from dbo.SaldoSimpanan where
        SaldoS_NamaAnggota_FK =" + ID, con);
        saldo =
        Convert.ToDouble(com.ExecuteScalar());
        com = new SqlCommand("select
        Nominal_Terbilang from dbo.Nominal_SW where Nominal_id=
        " + nominal, con);
        jumlah =
        Convert.ToDouble(com.ExecuteScalar());
        com = new SqlCommand("select SimpanW_id
        from dbo.Trans_SimpanWajib where SimpanW_NamaAnggota_FK=
        " + ID, con);
    }
}

```

```

        IDsimpanWajib
Convert.ToInt32(com.ExecuteScalar());
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
}
if (IDsimpanPokok != 0)//Apakah nasabah sudah
registrasi Simpanan Pokok?
{
    con.Open();
    com = new SqlCommand("select
SimpanP_NoRek      from dbo.Trans_SimpanPokok where
SimpanP_id= " + IDsimpanPokok, con);
    noRekDb = (string)com.ExecuteScalar();
    con.Close();
    if (IDsimpanWajib != 0)
    {
        con.Open();
        com = new SqlCommand("select
SimpanW_Tanggal  from dbo.Trans_SimpanWajib where
SimpanW_NamaAnggota_FK = " + ID, con);
        com.CommandType = CommandType.Text;

        string tglTerakhirSimpan = "";
        SqlDataReader      reader
        com.ExecuteReader();
        while (reader.Read())
        {
            tglTerakhirSimpan
            reader[0].ToString();
        }
        reader.Close();
        com.Dispose();
        con.Close();

        TglISWNext
Convert.ToDateTime(tglTerakhirSimpan);
        TglHarusSimpan
        TglISWNext.AddMonths(1);
        if (TglISW > TglHarusSimpan)
        {
            status = "Anda telah melakukan
Simpanan Wajib. Tgl terakhir menyimpan adalah " +
TglHarusSimpan.ToString("dd-MM-yyyy");
        }
        else { status = ""; }
    }
    else
    {
        TglHarusSimpan = TglISW.AddMonths(1);
        if (TglISW > TglHarusSimpan)
        {
            status = "Anda telah melakukan
Simpanan Wajib. Tgl terakhir menyimpan adalah " +
TglHarusSimpan.ToString("dd-MM-yyyy");
        }
        else { status = ""; }
    }
    saldo = saldo + jumlah;
}
else//jika simpan pokok belum memiliki data
{
    return RedirectToAction("CreateAwal",
new { status = 1 });
}
//mencari id Pengelola
try
{
    con.Open();
    com = new SqlCommand("Select
Pengelola_id from Pengelola where Pengelola_Nama = '" +
penerima + "'", con);
    int      idPengelola
Convert.ToInt32(com.ExecuteScalar());
    con.Close();
    ViewBag.Penerima = idPengelola;
}
catch (Exception x)
{
    throw x;
}
ViewBag.Status = status;
ViewBag.Kode = kode;
ViewBag>Nama = nama;
ViewBag.Nominal = nominal;
ViewBag.Tanggal = tanggal.ToString("yyyy-MM-
dd");
ViewBag.NoRek = noRekDb;

```

```

ViewBag.Saldo = saldo;
return View();
}
// POST: Trans_SimpanWajib/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"SimpanW_id,SimpanW_Kode_FK,SimpanW_NamaAnggota_FK,Simp
anW_NoRek,SimpanW_Nominal_FK,SimpanW_Tanggal,SimpanW_Pe
rnerima_FK,SimpanW_Saldo")]
Trans_SimpanWajib trans_SimpanWajib, String
nama, decimal saldo)
{
    if (ModelState.IsValid)
    {
        db.Trans_SimpanWajib.Add(trans_SimpanWajib);
        trans_SimpanWajib.SimpanW_Saldo = saldo;
        db.SaveChanges();
        try
        {
            con = new
SqlConnection(connectionString);
            con.Open();
            com = new SqlCommand("update
dbo.SaldoSimpanan set SaldoS_Jumlah = " + saldo + " where
SaldoS_NamaAnggota_FK = " + nama, con);
            com.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception x)
        {
            throw x;
        }
        return RedirectToAction("Details", new {
id = trans_SimpanWajib.SimpanW_id });
    }
    ViewBag.SimpanW_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanWajib.SimpanW_Kode_FK);
    ViewBag.SimpanW_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_SimpanWajib.SimpanW_NamaAnggota_FK);
    ViewBag.SimpanW_Nominal_FK = new
SelectList(db.Nominal_SW, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanWajib.SimpanW_Nominal_FK);
    ViewBag.SimpanW_Penerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama",
trans_SimpanWajib.SimpanW_Penerima_FK);
    return View(trans_SimpanWajib);
}
// GET: Trans_SimpanWajib/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return
new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Trans_SimpanWajib trans_SimpanWajib =
db.Trans_SimpanWajib.Find(id);
    if (trans_SimpanWajib == null)
    {
        return HttpNotFound();
    }
    ViewBag.SimpanW_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanWajib.SimpanW_Kode_FK);
    ViewBag.SimpanW_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_SimpanWajib.SimpanW_NamaAnggota_FK);
    ViewBag.SimpanW_Nominal_FK = new
SelectList(db.Nominal_SW, "Nominal_id",
"Nominal_Terbilang",
trans_SimpanWajib.SimpanW_Nominal_FK);
    ViewBag.SimpanW_Penerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama",
trans_SimpanWajib.SimpanW_Penerima_FK);
    return View(trans_SimpanWajib);
}
}

```

```

// POST: Trans_SimpanWajib/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"SimpanW_id,SimpanW_Kode_FK,SimpanW>NamaAnggota_FK,SimpanW_NoRek,SimpanW_Nominal_FK,SimpanW_Tanggal,SimpanW_Penerima_FK,SimpanW_Saldo")] Trans_SimpanWajib trans_SimpanWajib)
{
    if (ModelState.IsValid)
    {
        db.Entry(trans_SimpanWajib).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.SimpanW_Kode_FK = new SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi", trans_SimpanWajib.SimpanW_Kode_FK);
    ViewBag.SimpanW>NamaAnggota_FK = new SelectList(db.Anggotas, "Angg_ID", "Angg>Nama", trans_SimpanWajib.SimpanW>NamaAnggota_FK);
    ViewBag.SimpanW_Nominal_FK = new SelectList(db.Nominal_SW, "Nominal_id", "Nominal_Terbilang", trans_SimpanWajib.SimpanW_Nominal_FK);
    ViewBag.SimpanW_Penerima_FK = new SelectList(db.Pengelolas, "Pengelola_id", "Pengelola>Nama", trans_SimpanWajib.SimpanW_Penerima_FK);
    return View(trans_SimpanWajib);
}

// GET: Trans_SimpanWajib/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    Trans_SimpanWajib trans_SimpanWajib = db.Trans_SimpanWajib.Find(id);
    if (trans_SimpanWajib == null)
    {
        return HttpNotFound();
    }
    return View(trans_SimpanWajib);
}

// POST: Trans_SimpanWajib/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Trans_SimpanWajib trans_SimpanWajib = db.Trans_SimpanWajib.Find(id);
    db.Trans_SimpanWajib.Remove(trans_SimpanWajib);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

17. Source Coder

Trans_SimpananSukarela

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_SimpanSukarelaController : Controller
    {

```

```

private BMT_IMEntities1 db = new BMT_IMEntities1();
string connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
SqlConnection con;
SqlCommand com;

void Rupiah()
{
    System.Threading.Thread.CurrentThread.CurrentCulture = new System.Globalization.CultureInfo("id-ID");
    System.Threading.Thread.CurrentThread.CurrentUICulture = new System.Globalization.CultureInfo("id-ID");
}

// GET: Trans_SimpanSukarela
public ActionResult Index()
{
    Rupiah();
    var trans_SimpanSukarela = db.Trans_SimpanSukarela.Include(t => t.Anggota).Include(t => t.Pengelola);
    return View(trans_SimpanSukarela.ToList());
}

// GET: Trans_SimpanSukarela/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    Trans_SimpanSukarela trans_SimpanSukarela = db.Trans_SimpanSukarela.Find(id);
    if (trans_SimpanSukarela == null)
    {
        return HttpNotFound();
    }
    return View(trans_SimpanSukarela);
}

[Authorize]
public ActionResult CreateAwal()
{
    con = new SqlConnection(connectionString);
    string id = "";
    string userName = "";
    //mendapatkan pengelola
    try
    {
        con.Open();
        com = new SqlCommand("Select id from dbo.AspNetUsers where UserName = '" + User.Identity.Name + "'", con);
        id = (string)com.ExecuteScalar();
        com = new SqlCommand("Select Pengelola>Nama from Pengelola where Pengelola_UserName = '" + id + "'", con);
        userName = (string)com.ExecuteScalar();
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
    ViewBag.SimpanS_Kode_FK = new SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi");
    ViewBag.SimpanS>NamaAnggota_FK = new SelectList(db.Anggotas, "Angg_ID", "Angg>Nama");
    ViewBag.SimpanS_Penerima_FK = userName;
    return View();
}

[HttpGet]
// GET: Trans_SimpanSukarela/Create
public ActionResult Create(String kode, String nama, double nominal, DateTime tanggal, String penerima)
{
    string noRekDb = "";
    int IDsimpanPokok = 0;
    int IDsimpanWajib = 0;
    int ID = Convert.ToInt32(nama);
    con = new SqlConnection(connectionString);
    //mendapatkan id simpanan pokok dan wajib, dan jumlah saldo
    try
    {
        con.Open();

```

```

com = new SqlCommand("select SimpanW_id
from dbo.Trans_SimpanPokok where SimpanP_NamaAnggota_FK=
" + ID, con);
IDSimpanPokok =
Convert.ToInt32(com.ExecuteScalar());
com = new SqlCommand("select SimpanW_id
from dbo.Trans_SimpanWajib where SimpanW_NamaAnggota_FK=
" + ID, con);
IDSimpanWajib =
Convert.ToInt32(com.ExecuteScalar());
con.Close();
}
catch (Exception x)
{
throw x;
}
}
if (IDSimpanPokok != 0) //Apakah nasabah sudah
registrasi Simpanan Pokok?
{
if (IDSimpanWajib != 0)
{
con.Open();
com = new SqlCommand("select
SimpanP_NoRek
from dbo.Trans_SimpanPokok where
SimpanP_id= " + IDSimpanPokok, con);
noRekDb =
(string)com.ExecuteScalar();
con.Close();
}
else //jika simpan pokok belum memiliki data
{
return RedirectToAction("CreateAwal");
}
//mencari id Pengelola
try
{
con.Open();
com = new SqlCommand("Select
Pengelola_id from Pengelola where Pengelola_Nama = '" +
penerima + "'", con);
int idPengelola =
Convert.ToInt32(com.ExecuteScalar());
con.Close();
ViewBag.Penerima = idPengelola;
}
catch (Exception x)
{
throw x;
}
ViewBag.Kode = kode;
ViewBag>Nama = nama;
ViewBag.Nominal = nominal;
ViewBag.Tanggal = tanggal.ToString("yyyy-MM-
dd");
ViewBag.NoRek = noRekDb;
ViewBag.Saldo = nominal;
return View();
}
// POST: Trans_SimpanSukarela/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"SimpanS_id, SimpanS_Kode_FK, SimpanS_NamaAnggota_FK, Simp
anS_NoRek, SimpanS_Nominal, SimpanS_Tanggal, SimpanS_Perne
rima_FK, SimpanS_Saldo")] Trans_SimpanSukarela
trans_SimpanSukarela, String nama, decimal saldo)
{
if (ModelState.IsValid)
{
db.Trans_SimpanSukarela.Add(trans_SimpanSukarela);
trans_SimpanSukarela.SimpanS_Saldo =
saldo;
db.SaveChanges();
try
{
con = new
SqlConnection(connectionString);
con.Open();
com = new SqlCommand("update
dbo.SaldoSimpanan set SaldoS_Jumlah = " + saldo + " where
SaldoS_NamaAnggota_FK = " + nama, con);
com.ExecuteNonQuery();
con.Close();
}
}
}
}

```

```

}
catch (Exception x)
{
throw x;
}
return RedirectToAction("Index");
}
ViewBag.SimpanS_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanSukarela.SimpanS_Kode_FK);
ViewBag.SimpanS_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_SimpanSukarela.SimpanS_NamaAnggota_FK);
ViewBag.SimpanS_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengeola_Nama",
trans_SimpanSukarela.SimpanS_Pernerima_FK);
return View(trans_SimpanSukarela);
}
// GET: Trans_SimpanSukarela/Edit/5
public ActionResult Edit(int? id)
{
if (id == null)
{
return new
 HttpStatusCodeResult(HttpStatusCode.BadRequest);
}
Trans_SimpanSukarela trans_SimpanSukarela =
db.Trans_SimpanSukarela.Find(id);
if (trans_SimpanSukarela == null)
{
return HttpNotFound();
}
ViewBag.SimpanS_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanSukarela.SimpanS_Kode_FK);
ViewBag.SimpanS_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_SimpanSukarela.SimpanS_NamaAnggota_FK);
ViewBag.SimpanS_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama",
trans_SimpanSukarela.SimpanS_Pernerima_FK);
return View(trans_SimpanSukarela);
}
// POST: Trans_SimpanSukarela/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"SimpanS_id, SimpanS_Kode_FK, SimpanS_NamaAnggota_FK, Simp
anS_NoRek, SimpanS_Nominal, SimpanS_Tanggal, SimpanS_Perne
rima_FK, SimpanS_Saldo")] Trans_SimpanSukarela
trans_SimpanSukarela)
{
if (ModelState.IsValid)
{
db.Entry(trans_SimpanSukarela).State =
EntityState.Modified;
db.SaveChanges();
return RedirectToAction("Index");
}
ViewBag.SimpanS_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_SimpanSukarela.SimpanS_Kode_FK);
ViewBag.SimpanS_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_SimpanSukarela.SimpanS_NamaAnggota_FK);
ViewBag.SimpanS_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pergelola_Nama",
trans_SimpanSukarela.SimpanS_Pernerima_FK);
return View(trans_SimpanSukarela);
}
// GET: Trans_SimpanSukarela/Delete/5
public ActionResult Delete(int? id)
{
if (id == null)
{
return new
 HttpStatusCodeResult(HttpStatusCode.BadRequest);
}
Rupiah();
Trans_SimpanSukarela trans_SimpanSukarela =
db.Trans_SimpanSukarela.Find(id);
}
}

```

```

        if (trans_SimpanSukarela == null)
        {
            return HttpNotFound();
        }
        return View(trans_SimpanSukarela);
    }

    // POST: Trans_SimpanSukarela/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Trans_SimpanSukarela trans_SimpanSukarela =
        db.Trans_SimpanSukarela.Find(id);
        db.Trans_SimpanSukarela.Remove(trans_SimpanSukarela);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

18. Source Code Trans_Penarikan

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_PenarikanController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
        string connectionString =
        ConfigurationManager.ConnectionStrings["DefaultConnecti
        on"].ConnectionString;
        SqlConnection con;
        SqlCommand com;

        void Rupiah()
        {
            System.Threading.Thread.CurrentThread.CurrentCulture =
            new System.Globalization.CultureInfo("id-ID");

            System.Threading.Thread.CurrentThread.CurrentUICulture =
            new System.Globalization.CultureInfo("id-ID");
        }

        // GET: Trans_Penarikan
        public ActionResult Index()
        {
            Rupiah();
            var trans_Penarikan =
            db.Trans_Penarikan.Include(t => t.Anggota).Include(t =>
            t.Pengelola);
            return View(trans_Penarikan.ToList());
        }

        public ActionResult Export(int? id)
        {
            ReportDocument rd = new ReportDocument();

            rd.Load(Path.Combine(Server.MapPath("~/Report/Sim_Penar
            ik.rpt")));

            rd.SetDataSource(db.Trans_Penarikan.Select(p => new
            {
                Penarikan_id = p.Penarikan_id,
                Penarikan_Kode_FK =
                p.Trans_Kode.Kode_Transaksi,
                Penarikan_NamaAnggota_FK =
                p.Anggota.Angg_Nama,
                Penarikan_NoRek = p.Penarikan_NoRek,
                Penarikan_Nominal = p.Penarikan_Nominal,
                Penarikan_Tanggal = p.Penarikan_Tanggal,
                Penarikan_Penerima_FK =
                p.Pengelola.Pengelola_Nama,
                Penarikan_Saldo = p.Penarikan_Saldo
            }).Where(p => p.Penarikan_id ==
            id).ToList());
            Response.Buffer = false;
            Response.ClearContent();

```

```

            Response.ClearHeaders();
            Stream stream = rd.ExportToStream
            (CrystalDecisions.Shared.ExportFormatType.PortableDocFo
            rmat);
            stream.Seek(0, SeekOrigin.Begin);
            return File(stream, "application/pdf");
        }

        // GET: Trans_Penarikan/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Rupiah();
            Trans_Penarikan trans_Penarikan =
            db.Trans_Penarikan.Find(id);
            if (trans_Penarikan == null)
            {
                return HttpNotFound();
            }
            return View(trans_Penarikan);
        }

        [Authorize]
        public ActionResult CreateAwal(int? status)
        {
            con = new SqlConnection(connectionString);
            string id = "";
            string userName = "";
            string pesan = "";
            //mendapatkan pengelola
            try
            {
                con.Open();
                com = new SqlCommand("Select id from
                dbo.AspNetUsers where UserName = '" + User.Identity.Name
                + "'", con);
                id = (string)com.ExecuteScalar();
                com = new SqlCommand("Select
                Pengelola_Nama from Pengelola where Pengelola_UserName =
                '" + id + "'", con);
                userName = (string)com.ExecuteScalar();
                con.Close();
            }
            catch (Exception x)
            {
                throw x;
            }

            ViewBag.Penarikan_Kode_FK =
            new SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi");
            ViewBag.Penarikan_NamaAnggota_FK =
            new SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
            ViewBag.Penarikan_Penerima_FK = userName;
            if (status != null)
            {
                if (status == 1)
                {
                    pesan = "Anda belum melakukan
                    simpanan pokok !";
                }
                else if (status == 2)
                {
                    pesan = "Anda belum melakukan
                    simpanan wajib !";
                }
                else if (status == 3)
                {
                    pesan = "Saldo anda kurang dari Rp
                    100.000,00 !";
                }
                else if (status == 4)
                {
                    pesan = "Jumlah yang anda ambil
                    melebihi batas saldo !";
                }
            }
            ViewBag.Status = pesan;
            return View();
        }

        [HttpGet]
        // GET: Trans_Penarikan/Create
        public ActionResult Create(String kode, String
        nama, double nominal, DateTime tanggal, String penerima)
        {
            string noRekDb = "";
            double saldo = 0;
            int IDsimpanPokok = 0;
            int IDsimpanWajib = 0;
            int ID = Convert.ToInt32(nama);
            con = new SqlConnection(connectionString);
            //mendapatkan id simpanan pokok dan wajib,
            dan jumlah saldo
            try
            {
                con.Open();

```

```

        com = new SqlCommand("select SimpanP_id
from dbo.Trans_SimpanPokok where SimpanP_NamaAnggota_FK=
" + ID, con);
        IDsimpanPokok =
Convert.ToInt32(com.ExecuteScalar());
        com = new SqlCommand("select SimpanW_id
from dbo.Trans_SimpanWajib where SimpanW_NamaAnggota_FK=
" + ID, con);
        IDsimpanWajib =
Convert.ToInt32(com.ExecuteScalar());
        com = new SqlCommand("select
SaldoS_Jumlah from dbo.SaldoSimpanan where
SaldoS_NamaAnggota_FK = " + ID, con);
        saldo =
Convert.ToDouble(com.ExecuteScalar());
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
    if (IDsimpanPokok != 0)//Apakah nasabah sudah
registrasi Simpanan Pokok?
    {
        if (IDsimpanWajib != 0)
        {
            if (saldo > 100000)//jika saldo
melebihi batas minimal
            {
                if (saldo - nominal >
100000)//apabila uang yang diambil tidak lebih besar dari
saldo
                {
                    con.Open();
                    com = new SqlCommand("select
SimpanP_NoRek from dbo.Trans_SimpanPokok where
SimpanP_id= " + IDsimpanPokok, con);
                    noRekDb =
(string)com.ExecuteScalar();
                    con.Close();
                    saldo = saldo - nominal;
                }
                else
                {
                    return
RedirectToAction("CreateAwal", new { status = 4 });
                }
            }
            else{
                return
RedirectToAction("CreateAwal", new { status = 3 });
            }
            else
            {
                return
RedirectToAction("CreateAwal", new { status = 2 });
            }
        }
        //jika simpan pokok belum memiliki data
        else
        {
            return RedirectToAction("CreateAwal",
new { status = 1 });
        }
        //mencari id Pengelola
        try
        {
            con.Open();
            com = new SqlCommand("Select
Pengelola_id from Pengelola where Pengelola_Nama = '" +
penerima + "'", con);
            int idPengelola =
Convert.ToInt32(com.ExecuteScalar());
            con.Close();
            ViewBag.Penerima = idPengelola;
        }
        catch (Exception x)
        {
            throw x;
        }
        ViewBag.Kode = kode;
        ViewBag>Nama = nama;
        ViewBag.Nominal = nominal;
        ViewBag.Tanggal = tanggal.ToString("yyyy-MM-
dd");
        ViewBag.NoRek = noRekDb;
        ViewBag.Saldo = saldo;
        return View();
    }

    // POST: Trans_Penarikan/Create
    // To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]

```

```

public ActionResult Create([Bind(Include =
"Penarikan_id, Penarikan_Kode_FK, Penarikan_NamaAnggota_F
K, Penarikan_NoRek, Penarikan_Nominal, Penarikan_Tanggal, P
enarikan_Pernerima_FK, Penarikan_Saldo")]
Trans_Penarikan trans_Penarikan, String nama,
decimal saldo)
{
    con = new SqlConnection(connectionString);
    if (ModelState.IsValid)
    {
        db.Trans_Penarikan.Add(trans_Penarikan);
        trans_Penarikan.Penarikan_Saldo = saldo;
        db.SaveChanges();
        try
        {
            //memasukan data
            con.Open();
            //update saldo simpanan
            com = new SqlCommand("update
dbo.SaldoSimpanan set SaldoS_Jumlah = " + saldo + " where
SaldoS_NamaAnggota_FK = " + nama, con);
            com.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception x)
        {
            throw x;
        }
        return RedirectToAction("Details", new {
id = trans_Penarikan.Penarikan_id });
    }
    ViewBag.Penarikan_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_Penarikan.Penarikan_Kode_FK);
    ViewBag.Penarikan_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_Penarikan.Penarikan_NamaAnggota_FK);
    ViewBag.Penarikan_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama",
trans_Penarikan.Penarikan_Pernerima_FK);
    return View(trans_Penarikan);
}

// GET: Trans_Penarikan/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return
new
 HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Trans_Penarikan trans_Penarikan =
db.Trans_Penarikan.Find(id);
    if (trans_Penarikan == null)
    {
        return HttpNotFound();
    }
    ViewBag.Penarikan_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_Penarikan.Penarikan_Kode_FK);
    ViewBag.Penarikan_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_Penarikan.Penarikan_NamaAnggota_FK);
    ViewBag.Penarikan_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama",
trans_Penarikan.Penarikan_Pernerima_FK);
    return View(trans_Penarikan);
}

// POST: Trans_Penarikan/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Penarikan_id, Penarikan_Kode_FK, Penarikan_NamaAnggota_F
K, Penarikan_NoRek, Penarikan_Nominal, Penarikan_Tanggal, P
enarikan_Pernerima_FK, Penarikan_Saldo")] Trans_Penarikan
trans_Penarikan)
{
    if (ModelState.IsValid)
    {
        db.Entry(trans_Penarikan).State =
EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}

```

```

viewBag.Penarikan_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
trans_Penarikan.Penarikan_Kode_FK);
ViewBag.Penarikan_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_Penarikan.Penarikan_NamaAnggota_FK);
ViewBag.Penarikan_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama",
trans_Penarikan.Penarikan_Pernerima_FK);
return View(trans_Penarikan);
}

// GET: Trans_Penarikan/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    trans_Penarikan =
    db.Trans_Penarikan.Find(id);
    if (trans_Penarikan == null)
    {
        return HttpNotFound();
    }
    return View(trans_Penarikan);
}

// POST: Trans_Penarikan/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    trans_Penarikan =
    db.Trans_Penarikan.Find(id);
    db.Trans_Penarikan.Remove(trans_Penarikan);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

19. Source Code Trans_Pinjam

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_PinjamController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
        string connectionString =
        ConfigurationManager.ConnectionStrings["DefaultConnecti
on"].ConnectionString;
        SqlConnection con;
        SqlCommand com;

        void Rupiah()
        {
            System.Threading.Thread.CurrentThread.CurrentCulture =
            new System.Globalization.CultureInfo("id-ID");

            System.Threading.Thread.CurrentThread.CurrentUICulture =
            new System.Globalization.CultureInfo("id-ID");
        }

        // GET: Trans_Pinjam
        public ActionResult Index()
        {
            Rupiah();
            var trans_Pinjam = db.Trans_Pinjam.Include(t
=> t.Anggota).Include(t => t.Pengelola).Include(t =>
t.Nominal_P);
            return View(trans_Pinjam.ToList());
        }

        public ActionResult Export(int? id)
        {

```

```

reportDocument rd = new reportDocument();
rd.Load(Path.Combine(Server.MapPath("~/Report/Pinj_ rpt
")));
rd.SetDataSource(db.Trans_Pinjam.Select(p =>
new
{
    Pinjam_id = p.Pinjam_id,
    Pinjam_Kode_FK =
    p.Trans_Kode.Kode_Transaksi,
    Pinjam_NamaAnggota_FK =
    p.Anggota.Angg_Nama,
    Pinjam_NoRek = p.Pinjam_NoRek,
    Pinjam_Nominal_FK =
    p.Nominal_P.Nominal_Terbilang,
    Pinjam_Durasi = p.Pinjam_Durasi,
    Pinjam_AP = p.Pinjam_AP,
    Pinjam_BH = p.Pinjam_BH,
    Pinjam_ZK = p.Pinjam_ZK,
    Pinjam_AngsuranBulan =
    p.Pinjam_AngsuranBulan,
    Pinjam_Tanggal = p.Pinjam_Tanggal,
    Pinjam_Pernerima_FK =
    p.Pengelola.Pengelola_Nama,
    Pinjam_Saldo = p.Pinjam_Saldo
}).Where(p => p.Pinjam_id == id).ToList());
Response.Buffer = false;
Response.ClearContent();
Response.ClearHeaders();
Stream stream = rd.ExportToStream

(CrystalDecisions.Shared.ExportFormatType.PortableDocFo
rmat);
stream.Seek(0, SeekOrigin.Begin);
return File(stream, "application/pdf");
}

// GET: Trans_Pinjam/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    trans_Pinjam =
    db.Trans_Pinjam.Find(id);
    if (trans_Pinjam == null)
    {
        return HttpNotFound();
    }
    return View(trans_Pinjam);
}

[Authorize]
public ActionResult CreateAwal(int? status)
{
    con = new SqlConnection(connectionString);
    string id = "";
    string userName = "";
    string pesan = "";
    //mendapatkan pengelola
    try
    {
        con.Open();
        com = new SqlCommand("Select id from
dbo.AspNetUsers where UserName = '" + User.Identity.Name
+ "'", con);
        id = (string)com.ExecuteScalar();
        com = new SqlCommand("Select
Pengelola_Nama from Pengelola where Pengelola_UserName =
'" + id + "'", con);
        userName = (string)com.ExecuteScalar();
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
    ViewBag.Pinjam_Kode_FK = new
    SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi");
    ViewBag.Pinjam_NamaAnggota_FK = new
    SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
    ViewBag.Pinjam_Nominal_FK = new
    SelectList(db.Nominal_P, "Nominal_id",
    "Nominal_Terbilang");
    ViewBag.Pinjam_Pernerima_FK = userName;
    if (status != null)
    {
        if(status == 1)

```

```

simpanan pokok !";
    pesan = "Anda belum melakukan
    else
        pesan = "Anda masih mempunyai
pinjaman. Mohon dilunasi dahulu !";
    }
    ViewBag.Status = pesan;
    return View();
}
[HttpGet]
// GET: Trans_Pinjam/Create
public ActionResult Create(String kode, String
nama, double uang, String durasi, DateTime tanggal,
String penerima)
{
    string noRekDb = "";
    int ID = Convert.ToInt32(nama);
    int IDsimpanPokok = 0;
    int selama = 0;
    double AP = 0;
    double BH = 0;
    double ZK = 0;
    double angsuanBulan = 0;
    double saldoPinjaman = 0;
    double utang = 0;

    con = new SqlConnection(connectionString);
    con.Open();
    com = new SqlCommand("select SimpanP_id from
dbo.Trans_SimpanPokok where SimpanP_NamaAnggota_FK= " +
ID, con);
    IDsimpanPokok =
    Convert.ToInt32(com.ExecuteScalar());
    com = new SqlCommand("select SaldoP_jumlah
from dbo.SaldoPinjaman where SaldoP_NamaAnggota_FK = " +
ID, con);
    utang =
    Convert.ToDouble(com.ExecuteScalar());
    com = new SqlCommand("select
Nominal_Terbilang from dbo.Nominal_P where Nominal_id= "
+ uang, con);
    double uangs =
    Convert.ToDouble(com.ExecuteScalar());
    string jenis = uangs.ToString();
    con.Close();

    if (IDsimpanPokok != 0)
    {
        if (utang == 0)
        {
            con.Open();
            com = new SqlCommand("select
SimpanP_NoRek from dbo.Trans_SimpanPokok where
SimpanP_id= " + IDsimpanPokok, con);
            noRekDb =
            (string)com.ExecuteScalar();
            con.Close();

            switch (durasi)
            {
                case "1":
                    selama = 12;
                    switch (jenis)
                    {
                        case "250000":
                            AP = 20000;
                            BH = 3500;
                            ZK = 200;
                            break;
                        case "300000":
                            AP = 25000;
                            BH = 4200;
                            ZK = 200;
                            break;
                        case "350000":
                            AP = 29200;
                            BH = 5000;
                            ZK = 250;
                            break;
                        case "400000":
                            AP = 33000;
                            BH = 5600;
                            ZK = 300;
                            break;
                        case "450000":
                            AP = 37500;
                            BH = 6400;
                            ZK = 350;
                            break;
                        case "500000":
                            AP = 42000;
                    }
                }
            }
        }
    }
}

```

```

        BH = 7100;
        ZK = 350;
        break;
    case "550000":
        AP = 45800;
        BH = 7800;
        ZK = 400;
        break;
    case "600000":
        AP = 50000;
        BH = 8500;
        ZK = 450;
        break;
    case "700000":
        AP = 58000;
        BH = 9900;
        ZK = 500;
        break;
    case "750000":
        AP = 62500;
        BH = 10600;
        ZK = 550;
        break;
    case "800000":
        AP = 67000;
        BH = 11300;
        ZK = 600;
        break;
    case "850000":
        AP = 70800;
        BH = 12000;
        ZK = 600;
        break;
    case "900000":
        AP = 75000;
        BH = 12750;
        ZK = 650;
        break;
    case "950000":
        AP = 79200;
        BH = 13500;
        ZK = 700;
        break;
    case "1000000":
        AP = 83000;
        BH = 14100;
        ZK = 700;
        break;
    default:
        AP = 0;
        BH = 0;
        ZK = 0;
        break;
    }
    break;
case "2":
    selama = 24;
    switch (jenis)
    {
        case "250000":
            AP = 10500;
            BH = 1800;
            ZK = 100;
            break;
        case "300000":
            AP = 12500;
            BH = 2150;
            ZK = 100;
            break;
        case "350000":
            AP = 14600;
            BH = 2500;
            ZK = 125;
            break;
        case "400000":
            AP = 16700;
            BH = 2850;
            ZK = 150;
            break;
        case "450000":
            AP = 18750;
            BH = 3200;
            ZK = 200;
            break;
        case "500000":
            AP = 20800;
            BH = 3600;
            ZK = 200;
            break;
        case "550000":
            AP = 23000;
    }
}

```



```

        return View();
    }

    // POST: Trans_Pinjam/Create
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "Pinjam_id,Pinjam_Kode_FK,Pinjam_NamaAnggota_FK,Pinjam_
    NoRek,Pinjam_Nominal_FK,Pinjam_Durasi,Pinjam_AP,Pinjam_
    BH,Pinjam_ZK,Pinjam_AngsuranBulan,Pinjam_Tanggal,Pinjam_
    Pernerima_FK,Pinjam_Saldo")] Trans_Pinjam trans_Pinjam,
    String nama, String pengelola, decimal saldo)
    {
        if (ModelState.IsValid)
        {
            db.Trans_Pinjam.Add(trans_Pinjam);
            trans_Pinjam.Pinjam_Pernerima_FK =
            Convert.ToInt32(pengelola);
            trans_Pinjam.Pinjam_Saldo = saldo;
            db.SaveChanges();
            try
            {
                con = new
                SqlConnection(connectionString);
                con.Open();
                com = new SqlCommand("insert into
                dbo.SaldoPinjaman values(" + nama + ", " + saldo + ")",
                con);
                com.ExecuteNonQuery();
                con.Close();
            }
            catch (Exception x)
            {
                throw x;
            }
            return RedirectToAction("Details", new {
            id = trans_Pinjam.Pinjam_id });
        }
        ViewBag.Pinjam_Kode_FK = new
        SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
        trans_Pinjam.Pinjam_Kode_FK);
        ViewBag.Pinjam_NamaAnggota_FK = new
        SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
        trans_Pinjam.Pinjam_NamaAnggota_FK);
        ViewBag.Pinjam_Nominal_FK = new
        SelectList(db.Nominal_P, "Nominal_id",
        "Nominal_Terbilang", trans_Pinjam.Pinjam_Nominal_FK);
        ViewBag.Pinjam_Pernerima_FK = new
        SelectList(db.Pengelolas, "Pengelola_id",
        "Pengelola_Nama", trans_Pinjam.Pinjam_Pernerima_FK);
        return View(trans_Pinjam);
    }

    // GET: Trans_Pinjam/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Trans_Pinjam trans_Pinjam =
        db.Trans_Pinjam.Find(id);
        if (trans_Pinjam == null)
        {
            return HttpNotFound();
        }
        ViewBag.Pinjam_Kode_FK = new
        SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
        trans_Pinjam.Pinjam_Kode_FK);
        ViewBag.Pinjam_NamaAnggota_FK = new
        SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
        trans_Pinjam.Pinjam_NamaAnggota_FK);
        ViewBag.Pinjam_Nominal_FK = new SelectList(db.Nominal_P,
        "Nominal_id", "Nominal_Terbilang",
        trans_Pinjam.Pinjam_Nominal_FK);
        ViewBag.Pinjam_Pernerima_FK = new
        SelectList(db.Pengelolas, "Pengelola_id",
        "Pengelola_Nama", trans_Pinjam.Pinjam_Pernerima_FK);
        return View(trans_Pinjam);
    }

    // POST: Trans_Pinjam/Edit/5
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to, for

```

```

http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "Pinjam_id,Pinjam_Kode_FK,Pinjam_NamaAnggota_FK,Pinjam_
    NoRek,Pinjam_Nominal_FK,Pinjam_Durasi,Pinjam_AP,Pinjam_
    BH,Pinjam_ZK,Pinjam_AngsuranBulan,Pinjam_Tanggal,Pinjam_
    Pernerima_FK,Pinjam_Saldo")] Trans_Pinjam trans_Pinjam)
    {
        if (ModelState.IsValid)
        {
            db.Entry(trans_Pinjam).State =
            EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.Pinjam_Kode_FK = new
        SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
        trans_Pinjam.Pinjam_Kode_FK);
        ViewBag.Pinjam_NamaAnggota_FK = new
        SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
        trans_Pinjam.Pinjam_NamaAnggota_FK);
        ViewBag.Pinjam_Nominal_FK = new
        SelectList(db.Nominal_P, "Nominal_id",
        "Nominal_Terbilang", trans_Pinjam.Pinjam_Nominal_FK);
        ViewBag.Pinjam_Pernerima_FK = new
        SelectList(db.Pengelolas, "Pengelola_id",
        "Pengelola_Nama", trans_Pinjam.Pinjam_Pernerima_FK);
        return View(trans_Pinjam);
    }

    // GET: Trans_Pinjam/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Rupiah();
        Trans_Pinjam trans_Pinjam =
        db.Trans_Pinjam.Find(id);
        if (trans_Pinjam == null)
        {
            return HttpNotFound();
        }
        return View(trans_Pinjam);
    }

    // POST: Trans_Pinjam/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Trans_Pinjam trans_Pinjam =
        db.Trans_Pinjam.Find(id);
        db.Trans_Pinjam.Remove(trans_Pinjam);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

20. Source Code Trans_Angsuran

```

namespace SIBIT_INSANMADANI.Controllers
{
    public class Trans_AngsuranController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
        string connectionString =
        ConfigurationManager.ConnectionStrings["DefaultConnecti
        on"].ConnectionString;
        SqlConnection con;
        SqlCommand com;

        void Rupiah()
        {

```

```

System.Threading.Thread.CurrentThread.CurrentCulture =
new System.Globalization.CultureInfo("id-ID");

System.Threading.Thread.CurrentThread.CurrentUICulture =
new System.Globalization.CultureInfo("id-ID");
}
// GET: Trans_Angsuran
public ActionResult Index()
{
    Rupiah();
    var trans_Angsuran =
db.Trans_Angsuran.Include(t => t.Anggota).Include(t =>
t.Pengelola);
    return View(trans_Angsuran.ToList());
}

public ActionResult Export(int? id)
{
    ReportDocument rd = new ReportDocument();

rd.Load(Path.Combine(Server.MapPath("~/Report/Pinj_Angsur.rpt")));
rd.SetDataSource(db.Trans_Angsuran.Select(p => new
{
    Angsur_id = p.Angsur_id,
    Angsur_NamaAnggota_FK =
p.Anggota.Angg_Nama,
    Angsur_NoRek = p.Angsur_NoRek,
    Angsur_Jumlah = p.Angsur_Jumlah,
    Angsur_Ke = p.Angsur_Ke,
    Angsur_SS = p.Angsur_SS,
    Angsur_Tanggal = p.Angsur_Tanggal,
    Angsur_Penerima_FK =
p.Pengelola.Pengelola_Nama,
    Angsur_Saldo = p.Angsur_Saldo
}).Where(p => p.Angsur_id == id).ToList());
Response.Buffer = false;
Response.ClearContent();
Response.ClearHeaders();
Stream stream = rd.ExportToStream

(CrystalDecisions.Shared.ExportFormatType.PortableDocFormat);
    stream.Seek(0, SeekOrigin.Begin);
    return File(stream, "application/pdf");
}
// GET: Trans_Angsuran/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Rupiah();
    var trans_Angsuran =
db.Trans_Angsuran.Find(id);
    if (trans_Angsuran == null)
    {
        return HttpNotFound();
    }
    return View(trans_Angsuran);
}

[Authorize]
public ActionResult CreateAwal(int? status)
{
    con = new SqlConnection(connectionString);
    string id = "";
    string userName = "";
    string pesan = "";
    //mendapatkan pengelol
    try
    {
        con.Open();
        com = new SqlCommand("Select id from
dbo.AspNetUsers where UserName = " + User.Identity.Name
+ "'", con);
        id = (string)com.ExecuteScalar();
        com = new SqlCommand("Select
Pengelola_Nama from Pengelola where Pengelola_UserName =
'" + id + "'", con);
        userName = (string)com.ExecuteScalar();
        con.Close();
    }
    catch (Exception x)
    {
        throw x;
    }
}

```

```

ViewBag.Angsur_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
ViewBag.Angsur_Penerima_FK = userName;
if (status != null)
{
    if (status == 1)
        pesan = "Anda belum tidak mempunyai
pinjaman !";
    else
        pesan = "Pinjaman anda sudah lunas
!";
}
ViewBag.Status = pesan;
return View();
}

[HttpGet]
// GET: Trans_Angsuran/Create
public ActionResult Create(String nama, DateTime
tanggal, String penerima)
{
    string noRekDb = "";
    int ID = Convert.ToInt32(nama);
    int IDpinjam = 0;
    int IDangsur = 0;
    double angsuranBulan = 0;
    double saldoPinjaman = 0;
    int durasi = 0;
    string ke = "";
    string status = "";
    DateTime TglPinjam;
    DateTime TglHarusBayar;
    DateTime TglAngsur = tanggal;

    con = new SqlConnection(connectionString);
    con.Open();
    com = new SqlCommand("select Pinjam_id from
dbo.Trans_Pinjam where Pinjam_NamaAnggota_FK = " + ID,
con);
    IDpinjam =
Convert.ToInt32(com.ExecuteScalar());
    con.Close();

    if (IDpinjam != 0)
    {
        con.Open();
        com = new SqlCommand("select
SaldoP_Jumlah from dbo.SaldoPinjaman where
SaldoP_NamaAnggota_FK = " + ID, con);
        saldoPinjaman =
Convert.ToDouble(com.ExecuteScalar());
        com = new SqlCommand("select
Pinjam_NoRek from dbo.Trans_Pinjam where Pinjam_id = " +
IDpinjam, con);
        noRekDb = (string)com.ExecuteScalar();
        com = new SqlCommand("select
Pinjam_AngsuranBulan from dbo.Trans_Pinjam where
Pinjam_id= " + IDpinjam, con);
        angsuranBulan =
Convert.ToDouble(com.ExecuteScalar());
        com = new SqlCommand("select
Pinjam_Durasi from dbo.Trans_Pinjam where Pinjam_id= " +
IDpinjam, con);
        durasi =
Convert.ToInt32(com.ExecuteScalar());
        com = new SqlCommand("select Angsur_id
from dbo.Trans_Angsuran where Angsur_NamaAnggota_FK = "
+ ID, con);
        IDangsur =
Convert.ToInt32(com.ExecuteScalar());
        com = new SqlCommand("select
Pinjam_Tanggal from Trans_Pinjam where
Pinjam_NamaAnggota_FK = " + ID, con);
        TglPinjam =
Convert.ToDateTime(com.ExecuteScalar());
        con.Close();

        if (IDangsur != 0)
        {
            con.Open();
            com = new SqlCommand("select
Angsur_Ke from dbo.Trans_Angsuran where
Angsur_NamaAnggota_FK = " + ID, con);
            com.CommandType = CommandType.Text;
            string angsurKe = "";
            SqlDataReader reader =
com.ExecuteReader();
            while (reader.Read())
            {
                angsurKe = reader[0].ToString();
            }
        }
    }
}

```

```

        reader.Close();
        com.Dispose();
        con.Close();
        string          jumlah          =
    angsurKe.Substring(0, 2);
        int next = Convert.ToInt32(jumlah) +
    1;

        TglHarusBayar          =
    TglPinjam.AddMonths(next);
        if (TglAngsur > TglHarusBayar)
        {
            status = "Anda sudah telat
    membayar. Tgl terakhir membayar adalah " +
    TglHarusBayar.ToString("dd-MM-yyyy");
        }
        else { status = ""; }
        if (next == durasi)
        {
            ke = next.ToString() + " dari "
    + durasi + " (Lunas)";
        }
        else if (next > durasi)
        {
            return
    RedirectToAction("CreateAwal", new { status = 2 });
    //sudah lunas
        }
        else
        {
            ke = next.ToString() + " dari "
    + durasi;
        }
        }
        }
        else
        {
            ke = "1 dari "+durasi;

            TglHarusBayar          =
    TglPinjam.AddMonths(1);
            if (TglAngsur > TglHarusBayar)
            {
                status = "Anda sudah telat
    membayar. Tgl terakhir membayar adalah " +
    TglHarusBayar.ToString("dd-MM-yyyy");
            }
            else { status = ""; }
        }
        //mencari id Pengelola
        try
        {
            con.Open();
            com = new SqlCommand("Select
    Pengelola_id from Pengelola where Pengelola_Nama = " +
    penerima + "'", con);
            int idPengelola          =
    Convert.ToInt32(com.ExecuteScalar());
            con.Close();
            ViewBag.Penerima = idPengelola;
        }
        catch (Exception x)
        {
            throw x;
        }
        }
        }
        else
        {
            return RedirectToAction("CreateAwal",
    new { status = 1 });
        }
        ViewBag.Status = status;
        ViewBag.Nama = nama;
        ViewBag.NoRek = noRekDb;
        ViewBag.Nominal = angsuranBulan;
        ViewBag.Ke = ke;
        ViewBag.Tanggal = tanggal.ToString("yyyy-MM-
    dd");
        ViewBag.Saldo = saldoPinjaman;
        return View();
    }

    // POST: Trans_Angsuran/Create
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
    "Angsur_id,Angsur_NamaAnggota_FK,Angsur_NoRek,Angsur_Ju

```

```

    mian,Angsur_SS,Angsur_Tanggal,Angsur_Penerima
    a_FK,Angsur_Saldo"]
        Trans_Angsuran trans_Angsuran, String nama,
    decimal angsurTotal, decimal SS, DateTime tanggal, String
    pengelola, decimal saldo)
    {
        decimal AP, BH, ZK, saldoZkDB;
        string noRek = "";
        con = new SqlConnection(connectionString);
        try
        {
            con.Open();
            com = new SqlCommand("select
    Pinjam_NoRek from dbo.Trans_Pinjam where
    Pinjam_NamaAnggota_FK = " + nama, con);
            noRek = (string)com.ExecuteScalar();
            //Mendapatkan AP, BH, ZK
            com = new SqlCommand("select Pinjam_AP
    from dbo.Trans_Pinjam where Pinjam_NamaAnggota_FK = " +
    nama, con);
            AP          =
    Convert.ToDecimal(com.ExecuteScalar());
            com = new SqlCommand("select Pinjam_BH
    from dbo.Trans_Pinjam where Pinjam_NamaAnggota_FK = " +
    nama, con);
            BH          =
    Convert.ToDecimal(com.ExecuteScalar());
            com = new SqlCommand("select Pinjam_ZK
    from dbo.Trans_Pinjam where Pinjam_NamaAnggota_FK = " +
    nama, con);
            ZK          =
    Convert.ToDecimal(com.ExecuteScalar());

            decimal saldoAP = saldo - AP;
            decimal saldoBH = saldoAP - BH;
            decimal saldoZK = saldoBH - ZK;

            if (ModelState.IsValid)
            {
                db.Trans_Angsuran.Add(trans_Angsuran);

                trans_Angsuran.Angsur_NamaAnggota_FK
    Convert.ToInt32(nama);
                trans_Angsuran.Angsur_NoRek = noRek;
                trans_Angsuran.Angsur_Jumlah
    =
                angsurTotal;
                trans_Angsuran.Angsur_SS = SS;
                trans_Angsuran.Angsur_Saldo = saldo
    -
    (AP + BH + ZK);
                trans_Angsuran.Angsur_Penerima_FK
    =
    Convert.ToInt32(pengelola);
                db.SaveChanges();

                //memasukan data ke AP, BH, ZK,
                com = new SqlCommand("insert into
    Trans_AP values (" + 6 + ", " + nama + ", " + noRek +
    ", " + AP + ", " + tanggal + ", " + pengelola + ", "
    +
    saldoAP + ")", con);
                com.ExecuteNonQuery();
                com = new SqlCommand("insert into
    Trans_BH values (" + 7 + ", " + nama + ", " + noRek +
    ", " + BH + ", " + tanggal + ", " + pengelola + ", "
    +
    saldoBH + ")", con);
                com.ExecuteNonQuery();
                //memasukan data ke ZK dan Saldo

                Infq/Zakat-----
                -----
                com = new SqlCommand("select
    SaldoIF_Jumlah from SaldoInfq where
    SaldoIF_NamaAnggota_FK = " + nama + "'", con);
                saldoZkDB          =
    Convert.ToDecimal(com.ExecuteScalar());
                if (saldoZkDB != 0)
                {
                    ZK = saldoZkDB + ZK;
                    com = new SqlCommand("update
    dbo.SaldoInfq set SaldoIF_Jumlah = " + ZK + " where
    Sa'ldoIF_NamaAnggota_FK = " + nama, con);
                    com.ExecuteNonQuery();
                }
                else
                {
                    com = new SqlCommand("insert
    into dbo.SaldoInfq values (" + nama + ", " + ZK + ")",
    con);
                    com.ExecuteNonQuery();
                }
                com = new SqlCommand("insert into
    Trans_ZK values (" + 8 + ", " + nama + ", " + noRek +

```

```

    ", " + ZK + ", '" + tanggal + "', " + pengelola + ", "
    + saldoZK + ")", con);
        com.ExecuteNonQuery();
        //-----
        //-----
        //Memasukan data ke S.Sukarela
        com = new SqlCommand("insert into
Trans_SimpanSukarela values (" + 3 + ", " + nama + ", '"
+ noRek + "', " + SS + ", '" + tanggal + "', " + pengelola
+ ", " + SS + ")", con);
        com.ExecuteNonQuery();
        com = new SqlCommand("update
dbo.SaldoPinjaman set SaldoP_Jumlah = "+saldoZK+ " where
SaldoP_NamaAnggota_FK = "+nama, con);
        com.ExecuteNonQuery();
        con.Close();
        return RedirectToAction("Details",
new { id = trans_Angsuran.Angsur_id });
    }
    catch (Exception x)
    {
        throw x;
    }
    ViewBag.Angsur_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_Angsuran.Angsur_NamaAnggota_FK);
    ViewBag.Angsur_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama", trans_Angsuran.Angsur_Pernerima_FK);
    return View(trans_Angsuran);
}

// GET: Trans_Angsuran/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return
        HttpStatusCode.BadRequest;
    }
    Trans_Angsuran trans_Angsuran =
db.Trans_Angsuran.Find(id);
    if (trans_Angsuran == null)
    {
        return HttpNotFound();
    }
    ViewBag.Angsur_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_Angsuran.Angsur_NamaAnggota_FK);
    ViewBag.Angsur_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama", trans_Angsuran.Angsur_Pernerima_FK);
    return View(trans_Angsuran);
}

// POST: Trans_Angsuran/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"Angsur_id,Angsur_NamaAnggota_FK,Angsur_NoRek,Angsur_Ju
mlah,Angsur_Ke,Angsur_Tanggal,Angsur_Pernerima_FK,Angsu
r_Saldo")] Trans_Angsuran trans_Angsuran)
{
    if (ModelState.IsValid)
    {
        db.Entry(trans_Angsuran).State
        EntityState.Modified;
        db.SaveChanges()
        return RedirectToAction("Index");
    }
    ViewBag.Angsur_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
trans_Angsuran.Angsur_NamaAnggota_FK);
    ViewBag.Angsur_Pernerima_FK = new
SelectList(db.Pengelolas, "Pengelola_id",
"Pengelola_Nama", trans_Angsuran.Angsur_Pernerima_FK);
    return View(trans_Angsuran);
}

// GET: Trans_Angsuran/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {

```

```

        return
        HttpStatusCode.BadRequest;
    }
    Rupiah();
    Trans_Angsuran trans_Angsuran =
db.Trans_Angsuran.Find(id);
    if (trans_Angsuran == null)
    {
        return HttpNotFound();
    }
    return View(trans_Angsuran);
}

// POST: Trans_Angsuran/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Trans_Angsuran trans_Angsuran =
db.Trans_Angsuran.Find(id);
    db.Trans_Angsuran.Remove(trans_Angsuran);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

21. Source Code Trans_IFQ

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_IFQController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();
        string connectionString =
        ConfigurationManager.ConnectionStrings["DefaultConnecti
on"].ConnectionString;
        SqlConnection con;
        SqlCommand com;

        void Rupiah()
        {
            System.Threading.Thread.CurrentThread.CurrentCulture =
            new System.Globalization.CultureInfo("id-ID");

            System.Threading.Thread.CurrentThread.CurrentUICulture =
            new System.Globalization.CultureInfo("id-ID");
        }
        // GET: Trans_IFQ
        public ActionResult Index()
        {
            Rupiah();
            var trans_IFQ = db.Trans_IFQ.Include(t =>
t.Anggota).Include(t => t.Pengelola).Include(t =>
t.Trans_Kode);
            return View(trans_IFQ.ToList());
        }
        public ActionResult Export(int? id)
        {
            ReportDocument rd = new ReportDocument();

            rd.Load(Path.Combine(Server.MapPath("~/Report/Infaq.rpt
")));
            rd.SetDataSource(db.Trans_IFQ.Select(p =>
new
            {
                IFQ_id = p.IFQ_id,
                IFQ_kode_FK
                = p.Trans_Kode.Kode_Transaksi,
                IFQ_NamaAnggota_FK
                = p.Anggota.Angg_Nama,
                IFQ_Nominal = p.IFQ_Nominal,
                IFQ_Tanggal = p.IFQ_Tanggal,
                IFQ_NoRek = p.IFQ_NoRek,
                IFQ_Pernerima_FK
                = p.Pengelola.Pengelola_Nama,
                IFQ_Saldo = p.IFQ_Saldo
            }

```

```

        Response.Buffer = false;
        Response.ClearContent();
        Response.ClearHeaders();
        Stream stream = rd.ExportToStream

(CrystalDecisions.Shared.ExportFormatType.PortableDocFo
rmat);
        stream.Seek(0, SeekOrigin.Begin);
        return File(stream, "application/pdf");
    }
    // GET: Trans_IFQ/Details/5
    public ActionResult Details(int? id)
    {
        if (id == null)
        {
            return new
HttpStatusCodesResult(HttpStatusCode.BadRequest);
        }
        Rupiiah();
        Trans_IFQ trans_IFQ = db.Trans_IFQ.Find(id);
        if (trans_IFQ == null)
        {
            return HttpNotFound();
        }
        return View(trans_IFQ);
    }
    [Authorize]
    public ActionResult CreateAwal(int? status)
    {
        con = new SqlConnection(connectionString);
        string id = "";
        string userName = "";
        string pesan = "";
        //mendapatkan pengelola
        try
        {
            con.Open();
            com = new SqlCommand("Select id from
dbo.AspNetUsers where UserName = '" + User.Identity.Name
+ "'", con);
            id = (string)com.ExecuteScalar();
            com = new SqlCommand("Select
Pengelola_Nama from Pengelola where Pengelola_UserName =
'" + id + "'", con);
            userName = (string)com.ExecuteScalar();
            con.Close();
        }
        catch (Exception x)
        {
            throw x;
        }
        ViewBag.IFQ_Kode_FK = new
SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi");
        ViewBag.IFQ_NamaAnggota_FK = new
SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
        ViewBag.IFQ_Penerima_FK = userName;
        if (status != null)
        {
            pesan = "Anda belum melakukan simpanan
pokok !";
        }
        ViewBag.Status = pesan;
        return View();
    }
    // GET: Trans_IFQ/Create
    public ActionResult Create(String kode, String
nama, double nominal, DateTime tanggal, String penerima)
    {
        string noRekDb = "";
        int IDsimpanPokok = 0;
        int ID = Convert.ToInt32(nama);
        con = new SqlConnection(connectionString);
        //mendapatkan id simpanan pokok dan wajib, dan jumlah
saldo
        try
        {
            con.Open();
            com = new SqlCommand("select SimpanP_id
from dbo.Trans_SimpanPokok where SimpanP_NamaAnggota_FK=
'" + ID, con);
            IDsimpanPokok =
Convert.ToInt32(com.ExecuteScalar());
            con.Close();
        }
        catch (Exception x)
        {
            throw x;
        }
    }

```

```

registral Simpanan Pokok?
    {
        con.Open();
        com = new SqlCommand("select
SimpanP_NoRek from dbo.Trans_SimpanPokok where
SimpanP_id= '" + IDsimpanPokok, con);
        noRekDb = (string)com.ExecuteScalar();
        con.Close();
    }
    else//jika simpan pokok belum memiliki data
    {
        return RedirectToAction("CreateAwal",
new { status = 1 });
    }
    //mencari id Pengelola
    try
    {
        con.Open();
        com = new SqlCommand("Select
Pengelola_id from Pengelola where Pengelola_Nama = '" +
penerima + "'", con);
        int idPengelola =
Convert.ToInt32(com.ExecuteScalar());
        con.Close();
        ViewBag.Penerima = idPengelola;
    }
    catch (Exception x)
    {
        throw x;
    }
    ViewBag.Kode = kode;
    ViewBag>Nama = nama;
    ViewBag.Nominal = nominal;
    ViewBag.Tanggal = tanggal.ToString("yyyy-MM-
dd");
    ViewBag.NoRek = noRekDb;
    ViewBag.Saldo = nominal;
    return View();
}

// POST: Trans_IFQ/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details
see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"IFQ_id,IFQ_Kode_FK,IFQ_NamaAnggota_FK,IFQ_NoRek,IFQ_No
minal,IFQ_Tanggal,IFQ_Penerima_FK,IFQ_Saldo")]
Trans_IFQ trans_IFQ, String nama, decimal saldo)
    {
        decimal saldoDB;
        if (ModelState.IsValid)
        {
            try
            {
                con = new
SqlConnection(connectionString);
                con.Open();
                com = new SqlCommand("select
SaldoIF_Jumlah from SaldoInfaq where
SaldoIF_NamaAnggota_FK = '"+nama+"', con);
                saldoDB =
Convert.ToDecimal(com.ExecuteScalar());
                if(saldoDB != 0)
                {
                    saldo = saldoDB + saldo;
                    com = new SqlCommand("update
dbo.SaldoInfaq set SaldoIF_Jumlah = " + saldo + " where
SaldoIF_NamaAnggota_FK = " + nama, con);
                    com.ExecuteNonQuery();
                }
                else
                {
                    com = new SqlCommand("insert
into dbc.SaldoInfaq values ('"+nama+"','"+saldo+"')", con);
                    com.ExecuteNonQuery();
                }
                con.Close();
            }
            catch (Exception x)
            {
                throw x;
            }
            db.Trans_IFQ.Add(trans_IFQ);
            trans_IFQ.IFQ_Saldo = saldo;
            db.SaveChanges();
            return RedirectToAction("Details", new {
id = trans_IFQ.IFQ_id });
        }
    }

```

```

        ViewBag.IFQ_NamaAnggota_FK = new
        SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
        trans_IFQ.IFQ_NamaAnggota_FK);
        ViewBag.IFQ_Pernerima_FK = new
        SelectList(db.Pengelolas, "Pengelola_id",
        "Pengelola_Nama", trans_IFQ.IFQ_Pernerima_FK);
        ViewBag.IFQ_Kode_FK = new
        SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
        trans_IFQ.IFQ_Kode_FK);

        return View(trans_IFQ);
    }

    // GET: Trans_IFQ/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Trans_IFQ trans_IFQ = db.Trans_IFQ.Find(id);
        if (trans_IFQ == null)
        {
            return HttpNotFound();
        }
        ViewBag.IFQ_NamaAnggota_FK = new
        SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
        trans_IFQ.IFQ_NamaAnggota_FK);
        ViewBag.IFQ_Pernerima_FK = new
        SelectList(db.Pengelolas, "Pengelola_id",
        "Pengelola_Nama", trans_IFQ.IFQ_Pernerima_FK);
        ViewBag.IFQ_Kode_FK = new
        SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
        trans_IFQ.IFQ_Kode_FK);
        return View(trans_IFQ);
    }

    // POST: Trans_IFQ/Edit/5
    // To protect from overposting attacks, please
    // enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
    "IFQ_id,IFQ_Kode_FK,IFQ_NamaAnggota_FK,IFQ_NoRek,IFQ_No
    minal,IFQ_Tanggal,IFQ_Pernerima_FK,IFQ_Saldo")]
    Trans_IFQ trans_IFQ)
    {
        if (ModelState.IsValid)
        {
            db.Entry(trans_IFQ).State =
            EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        ViewBag.IFQ_NamaAnggota_FK = new
        SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
        trans_IFQ.IFQ_NamaAnggota_FK);
        ViewBag.IFQ_Pernerima_FK = new
        SelectList(db.Pengelolas, "Pengelola_id",
        "Pengelola_Nama", trans_IFQ.IFQ_Pernerima_FK);
        ViewBag.IFQ_Kode_FK = new
        SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
        trans_IFQ.IFQ_Kode_FK);
        return View(trans_IFQ);
    }

    // GET: Trans_IFQ/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return
            HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Rupiah();
        Trans_IFQ trans_IFQ = db.Trans_IFQ.Find(id);
        if (trans_IFQ == null)
        {
            return HttpNotFound();
        }
        return View(trans_IFQ);
    }

    // POST: Trans_IFQ/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)

```

```

        Trans_IFQ trans_IFQ = db.Trans_IFQ.Find(id);
        db.Trans_IFQ.Remove(trans_IFQ);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

22. Source Code Trans_ZK

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_ZKController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        void Rupiah()
        {
            System.Threading.Thread.CurrentThread.CurrentCulture =
            new System.Globalization.CultureInfo("id-ID");
            System.Threading.Thread.CurrentThread.CurrentUICulture =
            new System.Globalization.CultureInfo("id-ID");
        }

        // GET: Trans_ZK
        public ActionResult Index()
        {
            Rupiah();
            var trans_ZK = db.Trans_ZK.Include(t =>
            t.Anggota).Include(t => t.Pengelola).Include(t =>
            t.Trans_Kode);
            return View(trans_ZK.ToList());
        }

        // GET: Trans_ZK/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Rupiah();
            Trans_ZK trans_ZK = db.Trans_ZK.Find(id);
            if (trans_ZK == null)
            {
                return HttpNotFound();
            }
            return View(trans_ZK);
        }

        // GET: Trans_ZK/Create
        public ActionResult Create()
        {
            ViewBag.ZK_NamaAnggota_FK = new
            SelectList(db.Anggotas, "Angg_ID", "Angg_Nama");
            ViewBag.ZK_Pernerima_FK = new
            SelectList(db.Pengelolas, "Pengelola_id",
            "Pengelola_Nama");
            ViewBag.ZK_Kode_FK = new
            SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi");
            return View();
        }

        // POST: Trans_ZK/Create
        // To protect from overposting attacks, please
        // enable the specific properties you want to bind to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "ZK_id,ZK_Kode_FK,ZK_NamaAnggota_FK,ZK_NoRek,ZK_Nominal
        ,ZK_Tanggal,ZK_Pernerima_FK,ZK_Saldo")]
        Trans_ZK
        trans_ZK)
        {
            if (ModelState.IsValid)

```

```

        db.Trans_ZK.Add(trans_ZK);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.ZK_NamaAnggota_FK = new
    SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
    trans_ZK.ZK_NamaAnggota_FK);
    ViewBag.ZK_Pernerima_FK = new
    SelectList(db.Pengelolas, "Pengelola_id",
    "Pengelola_Nama", trans_ZK.ZK_Pernerima_FK);
    ViewBag.ZK_Kode_FK = new
    SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
    trans_ZK.ZK_Kode_FK);
    return View(trans_ZK);
}

// GET: Trans_ZK/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Trans_ZK trans_ZK = db.Trans_ZK.Find(id);
    if (trans_ZK == null)
    {
        return HttpNotFound();
    }
    ViewBag.ZK_NamaAnggota_FK = new
    SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
    trans_ZK.ZK_NamaAnggota_FK);
    ViewBag.ZK_Pernerima_FK = new
    SelectList(db.Pengelolas, "Pengelola_id",
    "Pengelola_Nama", trans_ZK.ZK_Pernerima_FK);
    ViewBag.ZK_Kode_FK = new
    SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
    trans_ZK.ZK_Kode_FK);
    return View(trans_ZK);
}

// POST: Trans_ZK/Edit/5
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"ZK_id,ZK_Kode_FK,ZK_NamaAnggota_FK,ZK_NoRek,ZK_Nominal
,ZK_Tanggal,ZK_Pernerima_FK,ZK_Saldo")] Trans_ZK
trans_ZK)
{
    if (ModelState.IsValid)
    {
        db.Entry(trans_ZK).State =
        EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.ZK_NamaAnggota_FK = new
    SelectList(db.Anggotas, "Angg_ID", "Angg_Nama",
    trans_ZK.ZK_NamaAnggota_FK);
    ViewBag.ZK_Pernerima_FK = new
    SelectList(db.Pengelolas, "Pengelola_id",
    "Pengelola_Nama", trans_ZK.ZK_Pernerima_FK);
    ViewBag.ZK_Kode_FK = new
    SelectList(db.Trans_Kode, "Kode_id", "Kode_Transaksi",
    trans_ZK.ZK_Kode_FK);
    return View(trans_ZK);
}

// GET: Trans_ZK/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Trans_ZK trans_ZK = db.Trans_ZK.Find(id);
    if (trans_ZK == null)
    {
        return HttpNotFound();
    }
    return View(trans_ZK);
}

// POST: Trans_ZK/Delete/5

```

```

[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Trans_ZK trans_ZK = db.Trans_ZK.Find(id);
    db.Trans_ZK.Remove(trans_ZK);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}
}

```

23. Source Code Trans_Kode

```

namespace SIBMT_INSANMADANI.Controllers
{
    public class Trans_KodeController : Controller
    {
        private BMT_IMEntities1 db = new
        BMT_IMEntities1();

        // GET: Trans_Kode
        public ActionResult Index()
        {
            return View(db.Trans_Kode.ToList());
        }

        // GET: Trans_Kode/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Trans_Kode trans_Kode =
            db.Trans_Kode.Find(id);
            if (trans_Kode == null)
            {
                return HttpNotFound();
            }
            return View(trans_Kode);
        }

        // GET: Trans_Kode/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Trans_Kode/Create
        // To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"Kode_id,Kode_Transaksi,Kode_JenisTransaksi")]
Trans_Kode trans_Kode)
{
    if (ModelState.IsValid)
    {
        db.Trans_Kode.Add(trans_Kode);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(trans_Kode);
}

// GET: Trans_Kode/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
}

```